UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Sistema Jargão - Um sistema para Análise de Base de Dados em Linguagem Natural.

Por: Ivan Rizzo Guilherme

Orientador: Prof. Dr. Armando Freitas Rocha

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas.

Campinas - 1996

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

G945s

Guilherme, Ivan Rizzo
Sistema Jargão: Um sistema para a análise de base de dados em linguagem natural / Ivan Rizzo Guilherme. -- Campinas, SP: [s.n.], 1996.

Orientador: Armando Freitas Rocha.
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Inteligência Artificial. 2. Processamento de Linguagem Natural (Computação). I. Rocha, Armando Freitas. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica. III. Título.

Este trabalho contou com o apoio financeiro do

Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPQ
e da
Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior - CAPES
Programa Institucional de Capacitação Docente - PICD

A idéia

De onde ela vem?! De que matéria bruta
vem essa luz que sobre as nebulosas
cai de incógnitas criptas misteriosas
como as estalactites de uma gruta?

Vem da psicogenética e alta luta
do feixe de moléculas nervosas
que, em desintegrações maravilhosas,
delibera, e depois, quer e executa!

Vem do encéfalo absconso que a constringe,
chega em seguida as cordas da laringe,
tísica, tênue, mínima, raquítica...

Quebra a força centrípeta que a amarra
Mas de repente, e quase morta, esbarra
no molambo da língua paralítica.

Augusto dos Anjos

Dedico este trabalho
à minha esposa Andréa
e meus pais Evan e Lurdes

Agradecimentos

Ao prof. Dr. Armando Freitas da Rocha pela paciência e incentivo, pelas oportunas e valiosas sugestões e também por sua segura orientação que certamente muito contribuiram para o denvolvimento deste trabalho.

À Universidade Estadual de Campinas.

À Universidade Estadual Paulista - UNESP, e em especial à chefia, professores e funcionários do Departamento de Estatística, Matemática Aplicada e Computacional - IGCE - UNESP, pelo irrestríto apoio.

Aos colegas do GIA - Grupo de Inteligência Artificial do DEMAC/IGCE/ UNESP - Rio Claro.

Aos Engenheiros da Petrobras, Kazuo Miura, Yutaka Irokawa, Ademar Sato e Antônio Patrício pelas valiosas sugestões dadas durante a utilização do Jargão.

Aos funcionários do Departamento de Fisiologia e Biofisica/IB/UNICAMP e do Departamento de Engenharia de Petróleo/FEM/UNICAMP pelo apoio e amizade.

Ao Prof. Edson Françoso pela colaboração.

Ao Sr. Majer Kaplan que indiretamente financiou este trabalho.

Aos colegas do EINA, Lais, Caco, Marcelo Theoto, Rodrigo, Cláudio e Roberto Zulli pela companhia, colaboração, amizade e apoio durante a nosso convívio.

À Adriane pela companhia, colaboração, pelas valiosas sugestões e também por sua preciosa revisão deste texto.

A todas as pessoas que direta ou indiretamente colaboração com o desenvolvimento deste trabalho.

Resumo

Neste trabalho são apresentados os aspectos teóricos e a implementação de um sistema computacional, denominado de Jargão, para a tarefa da análise de Base de Dados em Linguagem Natural. O sistema utiliza os conceitos de um novo modelo de neurônio, formalizado através dos conceitos de linguagem nebulosa, que suporta o processamento da linguagem natural. Este modelo de neurônio é utilizado na construção de redes neurais estruturadas hierarquicamente que permitem representar os complexos processos da linguagem. A geração das estruturas das redes neurais hierárquicas é feita de acordo com algum modelo lingüístico ou conceitual, que é especificado pelo usuário através de uma sintaxe. Neste trabalho é apresentada a geração dessas estrutura de acordo com uma estruturação conceitual do conhecimento. Neste sistema, o usuário é responsável pela especificação de um sintaxe nebulosa que codifica o seu conhecimento em cada nível hierárquico, e pela a análise das informações obtidas em cada nível das redes neurais.

Finalmente, são apresentadas as aplicações utilizando o sistema Jargão nas áreas de Aquisição do Conhecimento, Recuperação de Informação e Filtragem de Informação.

Abstract

Theoretical aspects and the implementation of a computational system, called Jargão, for task of analisys of data base in natural language is presented. The Jargão system uses concepts of a new model of neuron, formalized by fuzzy language concepts, to supports natural language processing. This model of neuron is used to build a hierarchically strutured neural net wich is able to do complex language processing. The generation of the structure of this hierarchical neural net is made according to linguistic or conceptual model, specified by the user. The user is responsable for specification of the fuzzy sintax which codifies their knowledge at each hierarchical level, and by interpretation of result.

Finally, aplications in the areas of: Knowledge Aquisition, Conceptual Information Retrievel and Information Filtering, are presented.

INDÍCE

CAPÍTULO 1

CAPÍTULO 2

CAPÍTULO 3

CAPÍTULO 4

CAPÍTULO 5

CAPÍTULO 6

Lista de Figuras

CAPÍTULO 1

INTRODUÇÃO

1.1 Domínio do Problema

Nos últimos anos, em virtude da evolução tecnológica na área da informática, tem-se criado condições para o desenvolvimento de computadores com maior capacidade de processamento e armazenamento das informações (CD, disco rígido, etc). Por outro lado, com o aumento da escala de produção, o preço dos equipamentos vem diminuindo, permitindo sua utilização em larga escala nas mais diversas atividades.

Com a evolução dos equipamentos e a expansão do mercado dos computadores, os programas de computadores tornaram-se também mais poderosos e de menor custo. Como conseqüência desta evolução, os sistemas de banco de dados e editores de textos tornaram-se mais acessíveis e têm sido utilizados na construção de base de dados em linguagem natural, com grande número de informações (CD com referências, base de dados com relatórios técnicos em alguma área profissional, prontuários médicos, etc).

Uma outra alteração profunda e bem mais recente tem sido causada pela evolução dos sistemas de comunicação entre computadores, provocando o aumento da troca eletrônica de informações entre as pessoas.

Como conseqüência desta evolução tecnológica, surgiram alguns problemas que até bem pouco tempo não existiam:

a) base de dados em linguagem natural (BDLN) com grande número de informações;

b) aumento do recebimento de informações eletrônicas pelas pessoas (correio eletrônico, jornais, comunicados entre departamentos de grandes empresas, etc).

Os problemas provenientes dessas novas situações consistem de um interessante campo de pesquisa em algumas áreas da Ciência da Computação. Na Ciência da Computação a área de Recuperação de Informação é que tradicionalmente trabalha no contexto de extração de informação de base de dados com grande volume de informações. Nesta área desenvolve-se métodos eficientes para a indexação das

informações da base de dados. A forma que é freqüentemente adotada na recuperação de informações de BDLN é através da utilização das palavras-chaves.

Por outro lado, os pesquisadores da Inteligência Artificial (mais especificamente pela sub-área de processamento de linguagem natural) estão há tempo desenvolvendo sistemas para: a recuperação conceitual de informação de textos; a extração de padrões de textos; e a análise de textos. Nos últimos anos, o conhecimento adquirido na manipulação de poucos textos, vem sendo utilizado no desenvolvimento de sistemas visando a análise de grandes BDLN. Recentemente, para a filtragem das informações eletrônicas recebidas pelas pessoas, e altamente influenciada pelas áreas de Recuperação da Informação e Inteligência Artificial, surgiu uma nova área, a de Filtragem de Informação.

Neste trabalho, enfoca-se a utilização de técnicas de Inteligência Artificial na manipulação da base de dados em linguagem natural com grande quantidade de textos.

As BDLN são geralmente criadas para armazenar as informações das ocorrências em alguma área específica de especialização (prontuários médicos, relatórios técnicos, etc). Em virtude da grande quantidade das informações da BDLN, torna-se muitas vezes necessária:

a) a criação de ferramentas que permitam a visualização e categorização das informações contidas nas BDLN;

b) a criação de mecanismos para a indexação e recuperação conceitual dos textos da BDLN;

c) ferramentas para a extração de padrões das BDLN; e

d) ferramentas para análise quantitativa e qualitativa das informações da BDLN.

É importante destacar que o conteúdo das BDLN é uma fonte importante de conhecimento. O desenvolvimento de técnicas automáticas para a extração deste conhecimento é uma importante linha de pesquisa na área de Inteligência Artificial. O conhecimento extraído é geralmente utilizado no desenvolvimento de sistemas baseados em conhecimento.

1.2 Inteligência Artificial

Na Ciência da Computação a área da Inteligência Artificial é caracterizada por tratar os aspectos computacionais relacionados com a inteligência humana. As principais áreas tratadas pela Inteligência Artificial são: visão, linguagem escrita e falada, raciocínio, etc.

A linguagem escrita tem um importante papel no desenvolvimento do aprendizado e na comunicação das pessoas. Em virtude da complexidade da linguagem, o seu estudo tem requerido o conhecimento de outras áreas além da lingüística, tais como, neurofisiologia, psicologia, lógica, etc.

Com o surgimento dos computadores e fazendo uso do desenvolvimento dos conhecimentos relacionados com a linguagem, tem se desenvolvido a área de lingüística computacional e a de processamento da linguagem natural.

1.2.1 Processamento da Linguagem Natural (PLN)

A área de processamento da linguagem natural trabalha com problemas como: interfaces em linguagem natural para sistemas computacionais, interpretação e geração de textos, tradução automática, análise automatizada de textos, recuperação inteligente de informação e filtragem de informação. O desenvolvimento desta área tem sido feito por duas sub-áreas distintas em suas concepções básicas:

a) os cognitivistas: assumem que a unidade básica do processamento é o símbolo e todas as regras para o processamento são definidas previamente. São fortemente influenciados pelos trabalhos de Chomsky ([CHOM57], [CHOM65]);

b) os conexionistas: assumem que a unidade básica do processamento é o neurônio. Os neurônios são agregados através das conexões sinápticas na construção de estruturas mais complexas (relações, regras). As relações são obtidas a partir de um conjunto de exemplos ([WALT85], [RUME86], [McCL86]).

Pesquisadores destas duas sub-áreas têm apontado as vantagens existentes na sua respectiva sub-área e as desvantagens na outra, o que tem causado grandes discussões entre eles. Somente recentemente tem surgido sistemas que têm buscado a incorporação das vantagens existentes em ambas as sub-áreas ([SHAS88], [MIIK91], [ROSE91]). Este trabalho está dentro desta filosofia de incorporar o conhecimento de ambas sub-áreas para aplicarmos na obtenção de informações contidas em textos.

Uma das principais controvérsias existentes entre os cognitivistas e conexionistas consiste na forma de organizar e desenvolver os sistemas inteligentes. Os cognitivistas, no desenvolvimento do sistema, acreditam ser necessário a completa codificação das informações (regras e o domínio) a serem processadas. Por outro lado, os conexionistas acreditam que as informações são produtos de um processo de aprendizado. Por exemplo, no processamento da linguagem natural, os cognitivistas seguem a concepção de Chomsky, onde todas as regras sintáticas estão previamente codificadas, ou seja, são inatas. Os conexionistas pensam exatamente o oposto. Durante o processo embriogênico são criadas as estruturas cerebrais que em fases claramente definidas, incorporam os estímulos recebidos do ambiente, que depois de organizadas passam a ser utilizadas.

Portanto, os conexionistas acreditam que as regras de utilização da linguagem são aprendidas, não sendo necessariamente predefinidas. Esta discussão tem reflexos sobre este trabalho, uma vez que, quanto mais o usuário interage com a BDLN, mais complexas são as regras sintáticas codificadas.

1.2.1.1 Abordagem Cognitivista

As abordagens cognitivistas no processamento da linguagem são fundamentalmente influenciadas pelos três componentes do processo de compreensão da linguagem natural: a análise sintática, a análise semântica e a análise pragmática. A análise sintática corresponde a transformação de uma frase em uma estrutura sintática que mostra como as palavras estão relacionadas. A análise semântica é feita considerando-se os significados associados aos elementos presentes na estrutura sintática. A análise pragmática é feita levando-se em consideração o uso da linguagem em um dado contexto. Os analisadores implementados nos sistemas de processamento da linguagem têm sido caracterizados por darem ênfase à análise sintática ou à semântica.

Uma das principais tarefas dos pesquisadores de PLN consiste em aprimorar os analisadores utilizados. Dentro deste contexto, um dos caminhos seguidos tem sido a utilização dos conceitos da estatística na construção e aperfeiçoamento dos analisadores.

A seguir são discutidos os sistemas de PLN de acordo com os aspectos predominantes nos analisadores. Assim, são apresentados os sistemas onde predominam a sintaxe, a semântica e analisadores que incorporam informações estatísticas.

A) Sistemas Baseados em Sintaxe

Os primeiros passos no processamento de linguagem natural no contexto cognitivista seguem dos trabalhos de Chomsky ([CHOM57], [CHOM65]), com a definição da teoria formal da linguagem, e através da sua utilização, foram construídos os primeiros sistemas computacionais para o processamento da linguagem natural. Desde então, têm sido proposto outros analisadores sintáticos como: Redes de Transição Aumentada (RTN) ([WOOD70]), Gramática de Cláusulas Definidas (GCD)([PERE80]), etc; que são caracterizados pela inclusão de mais conhecimento associado à análise sintática. Outro aspecto importante neste contexto foi o desenvolvimento, por pesquisadores da lingüística, de gramáticas mais complexas, como por exemplo: a gramática transformacional, a gramática de casos, a gramática sistêmica, etc. Como conseqüência deste avanço, um número considerável de sistemas utilizando esses formalismos teóricos foram desenvolvidos ([WINO72], [WOOD73]).

Os analisadores sintáticos são utilizados na maioria dos sistemas de processamento de linguagem. Entretanto, alguns problemas são importantes destacar:

a) os analisadores sintáticos requerem uma completa definição das regras sintáticas e também do ambiente no qual o sistema é utilizado. Como conseqüência, torna-se necessário o desenvolvimento de grandes bases de conhecimento contendo o conhecimento sintático e do contexto ([LENA90 et alli]). A medida em que a base aumenta, o desempenho do sistema torna-se menor. Esses sistemas são também conhecidos como sistemas de processamento de linguagem baseados em conhecimento;

b) os analisadores sintáticos geralmente operam sobre frases e não incorporam a interpretação semântica.

B) Sistemas Baseados em Semântica

A abordagem na qual privilegia-se a semântica ao invés da sintaxe, e a análise textual ao invés da frasal, foi proposta inicialmente por Shank ([SHAN75] e [SHAN82]) e seu grupo.

Os sistemas desenvolvidos dentro deste contexto operam sobre um ou mais textos, produzindo paráfrases e resumos. Em cada texto é verificada a existência das primitivas semânticas definidas no formalismo de representação do conhecimento denominada de Dependência Conceitual ([SHAN75]), e em seguida, as primitivas encontradas têm sua consistência analisada de acordo com estruturas denominadas Scripts ([SHAN75]) ou Frames ([MAUL91], [JACO90]). A maioria dos trabalhos desenvolvidos utilizando esses conceitos são criados para produzirem representações detalhadas de um número pequeno de textos. Esses conceitos estão sendo agora utilizados em sistemas para manipular grande volume de textos em tarefas de extração de dados, recuperação conceitual de informações ([MAUL91]), filtragem de informações ([RAM92]).

C) Sistemas Estatísticos

As técnicas estatísticas são utilizadas para a análise quantitativa dos dados. As técnicas de Inteligência Artificial e Estatística são aplicadas juntamente no desenvolvimento de algoritmos de aprendizado ([PARS89]). Os algoritmos de aprendizado são construídos para obter os padrões de informações contidos em um conjunto de dados e codificá-los em algum formalismo de representação de conhecimento. Alguns desses algoritmos têm sido adaptados para operarem sobre grandes bases de dados em linguagem natural, visando a extração do conhecimento

conceitual ou sintático. O conhecimento extraído é codificado em estruturas complexas de representação do conhecimento (árvores de decisão, frames) ([SODE94], [LI95]). Em seguida, o conhecimento conceitual é utilizado para direcionar a extração dos padrões numéricos ou simbólicos contido nos textos. O conhecimento sintático é utilizado como informação no aprimoramento do desempenho dos analisadores sintáticos.

1.2.1.2 Abordagem Conexionista

A abordagem conexionista voltou novamente a ganhar espaço na Inteligência Artificial na década de 80 ([McCL86], [RUME86] e [WALT85]). As arquiteturas conexionistas são caracterizadas pela capacidade de processamento numérico. No processamento da linguagem, a aplicação de técnicas conexionistas tem alcançado resultados expressivos em tarefas denominadas de baixo nível, como reconhecimento de caracteres, reconhecimento e produção da fala, modelagem do efeito do contexto na compreensão da linguagem natural, etc.

Nas tarefas consideradas de alto nível, como análise de textos, tradução, indexação conceitual, etc, os resultados alcançados não são tão expressivos quanto os da abordagem cognitivista. Isto porque, essas tarefas são mais complexas e, geralmente, requerem a especificação de níveis hierárquicos de conhecimento, não sendo possível codificá-lo em uma única arquitetura conexionista; e ainda, são caracterizadas pelo processamento simbólico e não numérico.

A estratégia adotada consiste na adoção das técnicas conexionistas em conjunto com as técnicas cognitivistas. Os sistemas que seguem esta estratégia são denominados de sistemas híbridos. A abordagem dos sistemas híbridos visa incorporar as vantagens encontradas em cada uma das áreas, como por exemplo, os eficientes mecanismos de aprendizagem dos conexionistas e a capacidade de manipulação das complexas estruturas de representação de conhecimento dos cognitivistas. Para a área conexionista, o principal aspecto positivo da utilização de abordagens híbridas é permitir a leitura do conteúdo aprendido pela rede neural, evitando desta forma as críticas feitas às redes neurais de serem caixas pretas.

Neste caminho de desenvolvimento de sistemas híbridos, duas vertentes são seguidas. A primeira consiste em utilizar os modelos conexionistas (modelo PDP, redes recorrentes) para construir os formalismos cognitivistas de representação de conhecimento (Scripts, Frames e Redes Semânticas) ([ROSE91], [MIIK91], [SHAS88], [JOHN92]), ou seja, as representações são codificadas nas conexões das redes. Na segunda, uma estrutura de representação de conhecimento é utilizada para descrever a topologia da rede neural ([MACH91]). O esquema de representação híbrida permite a

inferência associativa e a dedutiva. Este trabalho segue o primeiro caminho, pois utiliza um modelo conexionista que permite a construção de estruturas associadas com o processamento simbólico de alto nível.

Seguindo uma tendência da área cognitivista, os conexionistas têm como grande desafio o desenvolvimento de sistemas híbridos que operam sobre textos ([MIIK91], [JOHN92]). Utilizando o modelo conexionista PDP ([McCL86], [RUME86]), Mikkulainem ([MIIK91]) desenvolveu um sistema modular com capacidade de gerar as paráfrases de um dado texto. O texto é fornecido como o conjunto de treinamento, e os eventos e as relações de causa e efeito encontradas são incorporadas na estrutura dos módulos da rede na forma de um Script. O Script é utilizado na geração das paráfrases. O resultado produzido por este sistema é correspondente aos produzidos pelos sistemas cognitivistas. Neste contexto, a grande vantagem deste sistema está na capacidade de extrair dos textos as relações de causa e efeito que nos modelos cognitivistas necessitam ser codificadas pelo usuário. Por outro lado, a principal restrição deste sistema consiste da incapacidade em manipular situações não usuais que não tenham feito parte do conjunto de treinamento, uma vez que as informações codificadas na rede estão restritas ao conjunto de treinamento.

As técnicas conexionistas são muito pouco utilizadas para a manipulação de base de dados com grande volume de textos. A tendência natural é a evolução dos sistemas híbridos, que passariam da manipulação de alguns textos para grandes bases de dados em linguagem natural.

As técnicas conexionistas têm sido aplicadas no desenvolvimento de sistemas para a recuperação de informações de base de dados com grande volume de textos ([ROSE91]). Pelas características dos problemas de Recuperação de Informação, as técnicas conexionistas mostram-se muito atraentes.

1.3 Sistema para análise de BDLN

Neste trabalho, é apresentado o sistema, chamado de Jargão, que auxilia os usuários nas atividades de análise de base de dados em linguagem natural. A denominação está relacionada ao principal aspecto do sistema, que é tirar proveito das restrições do contexto imposta pelas especializações. Assim, o nível de especificação do contexto é dependente do conhecimento que o usuário dispõe do conteúdo das BDLN.

Um dos principais objetivos no desenvolvimento dos sistemas que operam sobre BDLN consiste da obtenção das informações contidas nos textos. Neste contexto, o nível de especificação da informação desejada pode ser considerado uma medida nebulosa, podendo variar de informações gerais a informações altamente especifica. O

nível de especificação da informação desejada está diretamente relacionada com o grau de informação sintática e semântica associado ao analisador utilizado. Assim, caso deseja-se uma especificação detalhada da informação contida nos textos, pode-se utilizar os conceitos presentes em sistemas sintáticos e semânticos apresentados na seção anterior. Evidentemente, o custo computacional é extremamente alto. As soluções adotadas consistem em criar sistemas com analisadores sintáticos em que a análise é parcial, e leva-se também em consideração, as informações conceituais e semânticas ([JACO93]).

No sistema Jargão, a análise sintática é feita através da gramática nebulosa. A gramática é criada pelo usuário de acordo com o seu conhecimento da BDLN ou com o grau de especificação do conhecimento a ser obtido da BDLN. Quanto mais simples for a gramática nebulosa especificada, mais redundante e parcial será a análise sintática. A medida em que a complexidade da gramática cresce, a redundância e a parcialidade da análise diminuem.

Na descrição, nas seções anteriores, das abordagens conexionista e cognitivista, mostrou-se que algumas delas têm sido aplicadas em ferramentas para operar sobre textos e recentemente têm sido expandidas para operar com BDLN, produzindo bons resultados. Naturalmente, alguns aspectos presentes nesses sistemas são relevantes, outros não. A tendência verificada na literatura tem sido o desenvolvimento de ferramentas que incorporem conceitos correspondentes aos aspectos positivos presentes nessas abordagens.

O sistema Jargão utiliza conceitos provenientes das abordagens conexionistas e cognitivistas, portanto consiste em uma abordagem híbrida. A topologia do modelo de rede neural utilizado é definida de acordo com a gramática nebulosa especificada e com o conteúdo da BDLN. As redes neurais podem acoplar-se em vários níveis hierárquicos, onde cada nível corresponde aos diversos níveis de hierarquia do conhecimento: sentenças, parágrafos e textos. Portanto, esta abordagem difere tanto das abordagens conexionistas quanto cognitivistas, que são fundamentadas nas noções da gramática clássica, e também na forma em que é definida e gerada a topologia da rede.

Os sistemas conexionistas e os sistemas estatísticos são caracterizados por possuírem mecanismos eficientes de aprendizado. No sistema Jargão serão incorporados dois mecanismos de aprendizado, um através do modelo conexionista utilizado, e outro denominado de aprendizado "por ouvir dizer", que consiste das informações fornecidas pelo usuário na interação com o sistema.

A freqüência de um conceito na BDLN é uma informação importante nos modelos estatísticos. No modelo conexionista utilizado a freqüência determina os pesos das conexões entre os neurônios, e é também utilizada no processo de seleção dos

módulos das redes. A freqüência de uma informação é utilizada como uma medida quantitativa do conteúdo da BDLN.

Os aspectos teóricos utilizados neste trabalho são apresentados no capítulo 2. Inicialmente descreve-se os conceitos básicos da Teoria Formal da Linguagem e da Linguagem Nebulosa que são utilizados nas definições e discussões de algumas das novas propostas apresentadas para conceituar uma Linguagem Nebulosa. A teoria conceitual que é apresentada permite a representação hierárquica do conhecimento. Em seguida, apresenta-se um modelo de processamento de neurônios que permite o processamento simbólico e numérico, fundamentados nas noções de transmissores, receptores e controladores (T^R>>C). Utilizando este modelo de neurônio e os conceitos de linguagem nebulosa apresentados descreve-se a estrutura e a geração da Rede Neural Evolutiva (RNE). As redes neurais serão utilizadas no Jargão para codificar o conhecimento conceitual hierárquico contido no conjunto de treinamento de acordo com uma linguagem nebulosa.

No capítulo 3 apresenta-se a descrição do sistema Jargão de acordo com os conceitos apresentados no capítulo 2. Apresenta-se a estrutura e os passos para a geração das Redes Neurais Evolutivas hierarquicamente organizadas: rede de conceitos primitivos; rede de conceitos complexos; e rede de teorias. Em seguida, são apresentadas as formas de análise das RNE de acordo com os seguintes tipos de conhecimento: específico ou conceitual. Após a análise, o conhecimento produzido corresponde à síntese da base de dados em linguagem natural analisada.

A descrição dos vários módulos que compõem o sistema e os aspectos relacionados com a implementação computacional são apresentadas no capítulo 4.

No capítulo 5 abordam-se as aplicações desenvolvidas utilizando-se o sistema Jargão como ferramenta. As aplicações apresentadas são caracterizadas por terem uma BDLN com grande quantidade de textos. Apresenta-se, em linhas gerais, uma metodologia para aquisição de conhecimento em BDLN utilizando-se o Jargão, e a seguir, apresenta-se a descrição das aplicações desenvolvidas nas áreas de Recuperação de Informação e Filtragem de Informação.

Finalmente, no capítulo 6 são apresentadas as conclusões.

CAPÍTULO 2

FUNDAMENTOS TEÓRICOS

2.1 Introdução

Muitos dos conceitos utilizados neste trabalho são provenientes do estudo da linguagem. Deve-se destacar que devido a: sua grande complexidade, existência de várias abordagens no seu estudo e sua multidisciplinaridade, os conceitos associados à linguagem, ainda hoje, provocam muitas discussões. Em virtude destes fatos, pode-se encontrar várias definições de linguagem. A definição aqui adotada é a dada por Chomsky ([CHOM57]): "Doravante considerarei uma linguagem como um conjunto (finito ou infinito) de sentenças, cada uma finita em comprimento e construída a partir de um conjunto finito de elementos".

A lingüística é o campo da ciência responsável pelo estudo da língua(gem). Na década de 50, a partir dos trabalhos de Chomsky, as pesquisas em lingüística tomaram um novo rumo e ganharam mais força. Com o surgimento dos computadores e fazendo uso do desenvolvimento dos conhecimentos relacionados com a linguagem, tem-se desenvolvido a área da lingüística computacional, e dentro dela, a de processamento da linguagem natural (PLN).

A lingüística computacional conecta a lingüística à Inteligência Artificial, com influência da psicologia e da filosofia. Uma das principais ferramentas da lingüística computacional tem sido a teoria formal da linguagem.

2.2 Teoria Formal da Linguagem

O trabalho inicial da sistematização e do estudo matemático da linguagem foi feito por Chomsky ([CHOM57]), que definiu a linguagem formalmente como um conjunto de cadeias de caracteres composto por símbolos de um dado vocabulário, que são combinados de acordo com regras gramaticais. O conjunto de cadeias de caracteres

corresponde ao conjunto de todas as sentenças possíveis, e que podem ser finitas ou infinitas. O vocabulário de símbolos corresponde a um alfabeto finito de símbolos ou de palavras.

Portanto, de acordo com Chomsky ([CHOM65]), para formalizar uma linguagem é necessário definir:

- as palavras do vocabulário, também conhecidas por símbolos terminais, cujo conjunto é denotado por V_T.

- As categorias sintáticas das quais as cadeias de palavras são derivadas. Estas categorias são chamadas de variáveis e seu conjunto é denotado pelo símbolo V_N.

- As relações existente entre cadeias particulares de símbolos não terminais. As relações são chamada de produções e seu conjunto denotado por P. Sejam as cadeias compostas por símbolos terminais e variáveis representadas por letras gregas minúsculas (α, β,...). Seja:

$V = V_T \cup V_N$,

V^* o conjunto de todas sentenças compostas por símbolos de V, e

$V^+ = V^* - \{\varepsilon\}$, onde ε representa a cadeia vazia.

As produções são da forma $\alpha \rightarrow \beta$, onde α é formada por símbolos em V^* e β em V^+.

- O símbolo inicial S, que causa a geração de todas as possíveis sentenças de acordo com as produções especificadas em P.

Desta forma, pode-se especificar uma linguagem através de uma gramática G que é denotada formalmente por (V_N, V_T, P, S). A linguagem reconhecida por G é denotada por L(G).

2.2.1 Tipos de Linguagens

As gramáticas definidas de acordo com a teoria formal da linguagem podem gerar 4 tipos de linguagens que são descritas a seguir:

Tipo 0: Linguagem Recursivamente Enumerável. As gramáticas que geram as linguagens deste tipo não requerem restrições quanto a forma das produções, sendo, portanto, a mais geral.

Tipo 1: Linguagem Sensível ao Contexto. As gramáticas que geram as linguagens deste tipo requerem produções da forma $\alpha A\beta \rightarrow \alpha\gamma\beta$, onde α e β são cadeias arbitrárias, incluindo a cadeia vazia, A é um símbolo não-terminal não-vazio (V_N), e γ é uma cadeia não-vazia sobre V^*. As restrições das produções fazem com que o lado direito das regras tenham o mesmo número ou mais símbolos que o lado esquerdo. Informalmente, dizemos que "A pode ser substituído por γ no contexto α, β".

Tipo 2: Linguagem Livre de Contexto. As gramáticas que geram as linguagens deste tipo requerem produções que tenham um único símbolo terminal à

esquerda da regra. Livre de Contexto significa que cada palavra na linguagem ocorre em concordância com as regras que não são dependentes do contexto em que as palavras são usadas.

Tipo 3: Linguagem Regular. As gramáticas que geram as linguagens deste tipo requerem produções da forma $A \rightarrow \alpha B$ ou $A \rightarrow \alpha$, onde, A e B são variáveis e α é um símbolo terminal ou vazio.

As definições das gramáticas seguem de uma gramática mais geral (tipo 0) para uma mais restrita (tipo 3). Para cada tipo de linguagem existe um autômato correspondente que pode processar a linguagem.

A teoria formal da linguagem tem dado sustentação para o desenvolvimento das linguagens de computação. Para o processamento da linguagem natural (PLN), os tipos definidos por Chomsky têm sido fundamentalmente utilizados na parte de análise sintática de frases. Para análise semântica os resultados não têm sido satisfatórios.

2.2.2 Derivação de Cadeias

As regras de produção são aplicadas sobre as cadeias de uma linguagem formal e este processo é denominado de derivação. A derivação da cadeia $\gamma\beta\delta$ a partir de uma cadeia da linguagem $\gamma\alpha\delta$ pela aplicação da regra $\alpha \rightarrow \beta$ é denotada por:

$$d(\alpha,\beta) = \gamma\alpha\delta \Rightarrow \gamma\beta\delta$$

Sejam $\alpha_1, \alpha_2, \ldots, \alpha_n, \beta$ cadeias em V^*, as seguintes produções:

$$\alpha_1 \rightarrow \alpha_2, \alpha_2 \rightarrow \alpha_3 \ldots, \alpha_n \rightarrow \beta$$

e uma cadeia da linguagem $\gamma\alpha_1\delta$. Podemos afirmar que existe um seqüência de derivações

$$d(\alpha_1,\beta)= \gamma\alpha_1\delta \Rightarrow \gamma\alpha_2\delta \ldots \gamma\alpha_{n\delta} \Rightarrow \gamma\beta\delta.$$

O processo de derivação envolve o seguintes passos:

1) casamento de padrões: ocorre quando o lado esquerdo de uma regra de produção é idêntica ao símbolo da cadeia processada.

2) Processo de reescrita: o símbolo do casamento de padrões é substituído na cadeia processada pelo símbolo do lado direito da regra de produção.

3) Aceitação: o grau de aceitação de uma seqüência de derivações é calculado de acordo com o casamento de padrões em cada derivação. Assim, o valor da aceitação é 1 se casamento de padrões ocorre e 0 caso contrário.

Uma linguagem formal L(G) é definida como um subconjunto de cadeias de símbolos gerados de acordo com as especificações feitas numa gramática G. Uma cadeia α gerada por G e pertencente à L(G) é chamada de fórmula bem formada de L(G) , se

todos os elementos de α pertencem a V_T. Em outras palavras, a cadeia α aceita pela linguagem L(G), suportada por G, tem apenas elementos pertencentes a V_T, e é obtida por derivações a partir de S. Reescrevendo, temos:

$$d(S,\alpha) = S \Rightarrow \alpha$$

onde: S é um símbolo inicial de G;

α é composta por símbolos de V_T.

$\Rightarrow$ representa as derivações provocadas pela aplicação de produções de P.

Portanto, a linguagem gerada por uma gramática G é dada por:
$L(G) = \{\alpha |$ α é composta somente por símbolos terminais e cadeias derivadas a partir de S (símbolo inicial)$\}$.

O valor de aceitação de todas as cadeias $\alpha \in L(G)$ é igual a 1. O valor de aceitação associado às cadeias que não pertence a linguagem é 0.

2.3 Teoria Formal Nebulosa da Linguagem

2.3.1 Gramática Nebulosa

A gramática nebulosa ([MIZU73 et all]) é uma generalização da gramática formal, onde temos associado à cada produção, o valor de aplicação da produção. Portanto, uma gramática nebulosa é definida por:

$$G_n = (V_N, V_T, P, S, f)$$

onde:

(i) V_N é o vocabulário dos símbolos não-terminais;
(ii) V_T é o vocabulário dos símbolos terminais;
(iii) S representa subconjunto de símbolos iniciais em V_N;
(iv) P é um conjunto finito de produções da forma:

$$i: \alpha \rightarrow \beta(\mu_i)$$

Seja $V = V_T \cup V_N$, V^* o conjunto de todas sentenças compostas dos símbolos de V, e $V^+ = V^* - \{\varepsilon\}$, onde ε representa a cadeia vazia. Nas produções, α é formada por símbolos em V^+ e β em V^*. A cada produção é associado um rótulo *i*. O conjunto dos rótulos é denotado por I. Assim, o grau de aplicação associado à produção *i* é denotado por μ_i.

(v) μ é uma função de pertinência de

$$\mu: I \rightarrow [0,1]$$

que representa o grau de aplicação μ_i associado a regras produção com o rótulo $i \in I$.

2.3.2 Derivação na Linguagem Nebulosa

O processo de derivação é uma das formas de descrever a geração das sentenças de uma linguagem nebulosa ($L(G_n)$) de acordo com uma gramática nebulosa (G_n). A derivação em linguagens nebulosas ([MIZU73 et all]) é feita da seguinte forma: seja r uma produção $s_i \rightarrow s_j(\mu_r)$ em P, e α e β cadeias em V^*, então:

$$d(s_i, s_j) = \alpha s_i \beta \xrightarrow{\mu_r} \alpha s_j \beta$$

e $\alpha s_j \beta$ é dita ser derivada diretamente de $\alpha s_i \beta$ com grau μ_r pela produção r.

Sejam $\alpha_1, \alpha_2, \ldots, \alpha_m$ cadeias em V^*, e

$$\alpha_0 \xrightarrow{\mu r_1} \alpha_1,\ \alpha_1 \xrightarrow{\mu r_2} \alpha_2,\ \ldots,\ \alpha_{m-1} \xrightarrow{\mu r_m} \alpha_m$$

produções $r_1, r_2, \ldots, r_m$. Dada uma cadeia $u\alpha_0 v$, ao serem aplicadas as produções, ocorre uma seqüência de derivações:

$$d(\alpha_0, \alpha_m) = u\alpha_0 v \xrightarrow{\mu r_1} u\alpha_1 v \xrightarrow{\mu r_2} u\alpha_2 v \xrightarrow{\mu r_3} \ldots u\alpha_{m-1} v \xrightarrow{\mu r_m} u\alpha_m v$$

que é denominada de cadeia de derivação nebulosa com comprimento m de α_0 para α_m, através das regras de produções nebulosas $r_1 \ldots r_m$.

O processo de derivação na gramática nebulosa mostrado acima é mais complexo que o da gramática clássica (seção 2.2.2), porque na gramática nebulosa é especificada para cada regra, o grau de aplicação μ. Entretanto, outros aspectos relacionados à derivação nebulosa podem ser destacados. Assim, dado: uma regra nebulosa $s_i \rightarrow s_j(\mu_r)$; α a cadeia corrente processada; e s_k uma subcadeia de α. O processo de derivação envolve os seguintes passos:

a) casamento de padrões: ocorre quando a cadeia s_i do lado esquerdo da regra de produção acopla-se com a subcadeia s_k da cadeia processada. O acoplamento $\mu_c(s_i,s_k)$ é medido em $(V^*)^N$, onde N é o comprimento da menor cadeia, tal que:

$$\mu_c: (V^*)^N \rightarrow [0,1]$$

Desta forma, o acoplamento entre as cadeias pode ser parcial, sendo o valor associado ao acoplamento dado por:

$\mu_c: (s_i,s_k) \rightarrow 1$; se a cadeia s_i tende a ser igual a s_k;

$\mu_c: (s_i,s_k) \rightarrow 0$; caso contrário.

Pode-se ainda ser definido um limiar (θ), sendo o acoplamento aceito quando atingir um valor acima do limiar:

$$\mu_c: (V^*)^N \rightarrow [0,1] \text{ e } \mu_c(s_i,s_k) > \theta, \text{ e } \theta \in [0,1];$$

b) processo de reescrita: a subcadeia s_k da cadeia processada é substituída na cadeia processada pela cadeia (s_j) do lado direito da regra de produção nebulosa;

c) grau de aceitação: o grau de aceitação (μ_d) produzido pela derivação $d(s_i,s_k)$ é calculado em função do casamento de padrões da derivação ($\mu_c(s_i,s_k)$) e do grau associado à regra de produção (μ_r). Portanto,

$$\mu_d(s_i,s_k) = (\mu_c(s_i,s_k) \wedge \mu_r)$$

onde: $\wedge$ é um operador nebuloso correspondente a uma norma triangular (denominada de norma-t). A definição de norma-t é encontrada em [ZIME91].

As cadeias geradas por uma gramática nebulosa G_n correspondem a uma linguagem nebulosa $L(G_n)$ e são construídas por uma cadeia de derivação iniciada por um símbolo $s_0 \in S$, e que proporcione à cadeia $s_t \in L(G_n)$ somente símbolos pertencentes a V_T,

$$d(s_0,s_t) = \alpha s_0 \beta \xrightarrow{\mu_r} \alpha s_1 \beta \ldots \alpha s_n \beta \xrightarrow{\mu_r} s_t$$

e o grau de aceitação final é feito a partir do cálculo de cada uma das derivações parciais:

$\mu_d(s_0, s_k) = (\mu_c(s_0,s_k) \wedge \mu_{r2})$; /* Regra 2
$\mu_d(s_1, s_k) = (\mu_c(s_1,s_k) \wedge \mu_{r4})$; /* Regra 4
...
$\mu_d(s_n, s_k) = (\mu_c(s_n,s_k) \wedge \mu_{r1})$. /* Regra 1

Assim, o valor da aceitação final é calculado pela agregação dos valores obtidos das derivações parciais:

$$\mu_d(s_0, s_t) = \mu_d(s_0,s_k) \vee \mu_d(s_1,s_k) \vee \ldots \vee \mu_d(s_n,s_k)$$

$$\mu_d(s_0, s_t) = \bigvee_{i=1}^{n} (\mu_d(s_i,s_k)) \text{ e } \mu_d(s_0, s_t) \in [0,1]$$

que corresponde ao máximo ($\vee$) valor do grau de aceitação das n derivações.

Portanto, cada uma das cadeias s_t geradas de acordo com G_n tem um valor de aceitação associado ($\mu_d(s_0,s_t)$), que corresponde à pertinência de s_t em $L(G_n)$. O valor de pertinência corresponde à medida de quanto apropriada é a cadeia na linguagem.

2.3.3 Ambigüidade na Linguagem Nebulosa

As linguagens nebulosas exibem propriedades distintas das existentes na linguagem formal clássica. Isto porque, dado uma cadeia $s_i \in V^*$, podem existir muitas regras em P com a parte esquerda $s_k \in V^*$, tal que o casamento de padrões ocorra ($\mu_c(s_i,s_k) > 0$). Isto significa que muitas seqüências de derivações podem existir, permitindo a reescrita de s_i em diferentes $s_j \in V^*$.

Conseqüentemente, para a geração das cadeias de $L(G_n)$, de acordo com a gramática nebulosa G_n e composta por símbolos de V_T, pode-se requerer um número maior de cadeias de derivações que as da linguagem formal clássica. Como a ambigüidade na geração das cadeias de $L(G_n)$ está relacionada à quantidade de cadeias de derivações iniciadas por algum $s_0 \in S$ e que proporcione à $s_t \in L(G_n)$ somente símbolos pertencentes a V_T, pode-se afirmar que as linguagens geradas por gramáticas nebulosas são naturalmente ambíguas. Para reduzir a quantidade de cadeias de derivações e conseqüentemente a ambigüidade, algumas medidas especiais podem ser tomadas. A forma comumente adotada tem sido a utilização do limiar de aceitação ([MIZU73 et all]). Nela, as derivações são permitidas somente quando o grau de ativação resultante está acima do limiar. Como conseqüência, as cadeias de derivação são geradas com o grau de aceitação acima do limiar.

Em ([ROCH95 et alli]) uma outra forma para tratar a ambigüidade é apresentada. Nesta abordagem leva-se também em consideração a quantidade das cadeias no processo de derivação. Assim, na ativação da regra $\alpha s_k \beta \rightarrow \alpha s_j \beta$ para reescrita de s_i em s_j no processo de derivação $d(s_0,...s_i, s_j,...,s_t)$ da cadeia $s_t \in L(G_n)$, é também considerada a quantidade disponível de $A(s_i)$, $A(s_k)$ de s_i, $s_k \in V^*$. Entretanto, ao ser adotada esta estratégia deve-se obedecer, ao menos, restrições sobre os seguintes aspectos:

a) o número total $A(s_i)$ de cópias disponíveis de $s_i \in V^*$: pelo menos uma cópia de uma cadeia contendo $s_i \in V^*$ deve estar disponível para disparar cada possível cadeia de derivação $d(s_i,s_j)$, suportada por regras de reescrita do tipo $\alpha s_k \beta \rightarrow \alpha s_j \beta$ e tendo $\mu_c(s_i,s_k) > 0$;

b) o número total $A(s_k)$ de cópias de $s_k \in V^*$: ao menos uma cópia da cadeia $\alpha s_k \beta$ tem que estar disponível para permitir que a regra $\alpha s_k \beta \rightarrow \alpha s_j \beta$ seja usada, ocorrendo evidentemente, $\mu_c(s_i,s_k) > 0$;

c) o número atual $a(\alpha s_i \beta)$ de cópias ativadas da sentença $\alpha s_i \beta$ para ser processada: ao menos uma cópia da cadeia $\alpha s_i \beta$ tem que ser selecionada para ser reescrita.

Essas restrições implicam que não somente a qualidade do símbolo é levada em consideração no processamento nebuloso da linguagem, mas também a quantidade deve representar um importante papel na determinação da cadeia a ser utilizada no processo

de reescrita de uma sentença nebulosa. Esta abordagem contrasta com a abordagem clássica que não leva em consideração a quantidade no processo de cópia.

d) O número limite de elementos de V_T nas cadeias $s_t \in L(G_n)$: o limite do comprimento ($T(s_t)$) das cadeias $s_t \in L(G_n)$ pode ser especificado, reduzindo desta forma, o número de cadeias de derivações produzidas. Em outras palavras temos:

$$T(s_t) < l$$

onde, a especificação do valor l corresponde a um dos parâmetros que determinam o grau de ambigüidade ($g(L(G_n))$) da $L(G_n)$.

e) O número de elementos em V_T: o número de produção do tipo $s_k \rightarrow s_i$, onde $s_k \in V^*$ e $s_i \in V_T$. A restrição dos símbolos terminais é para evitar a explosão combinatória causada por dicionários complexos.

f) A prioridade de acoplamento: quando não existe quantidade suficiente de cópias disponíveis de $s_i \in V^*$ para disparar todas as derivações $d(s_i, s_j)$, então a prioridade deve ser dada àquelas regras $\alpha s_k \beta \rightarrow \alpha s_j \beta$ que exibem maior valor de $\mu_c(s_i, s_k)$. Outro aspecto a ser considerado é que a prioridade deve ser dada àquelas regras suportadas por $s_k \in V^*$ que tenham o maior número $A(s_k)$ de cópias disponíveis.

A possibilidade $\rho(d(s_i,s_j))$ do encadeamento da derivação $d(s_i, s_j)$, quando apenas uma regra $\alpha s_k \beta \rightarrow \alpha s_j \beta$, $\mu_c(s_i, s_k) > 0$ possibilita a reescrita de s_i em $s_j \in V^*$, é dependente da $A(s_i)$, $A(s_k)$ e $\mu_c(s_i,s_k)$, e é dada por:

$$\rho(d(s_i,s_j)) = f(A(s_i), A(s_k), \mu_c(s_i,s_k))$$

$$f: A(s_i) \times A(s_k) \times \mu_c(s_i,s_k) \rightarrow [0,1]$$

Nas situações onde n regras possibilitam a reescrita de s_i em s_j, a possibilidade $\rho(d(s_i,s_j))$ é dada por:

$$\rho(d(s_i,s_j)) = \max\ (f(A(s_i), A(s_1), \mu_c(s_i, s_1)), \text{ regra 1:}$$

$$f(A(s_i), A(s_2), \mu_c(s_i, s_2)), \quad \text{regra 2:}$$

$$\ldots$$

$$f(A(s_i), A(s_n), \mu_c(s_i, s_n))) \quad \text{regra n:}$$

$$\rho(d(s_i, s_j)) = \max_{z=1}^{n} (f(A(s_i), A(s_z), \mu_c(s_i, s_z)))$$

A quantidade $a(\alpha s_j \beta)$ de cópias $s_j \in V^*$ produzidas (ativadas) por $s_i \in V^*$ pode ser assumida dependente do numero de cópias correntes ($a(\alpha s_i \beta)$), do número de derivações ($d(s_i,s_j)$) disparadas e da possibilidade de cada uma dessas cadeias de derivações serem utilizadas por $s_i \in V^*$. A quantidade de cópias produzidas é limitada de acordo com um valor máximo A_{max}. Este valor é especificado de acordo com o

contexto e visa limitar o número de copias produzidas. Assim, utiliza-se este valor para normalizar o valor de copias da cadeia corrente ($a(\alpha s_i \beta)$). Desta forma, o valor normalizado do número corrente de cópias é dado por:

$$\bar{a}(\alpha s_i \beta) = \begin{cases} a(\alpha s_i \beta) \ / \ A_{max} & \text{se } a(\alpha s_i \beta) \leq A_{max} \\ 1 & \text{caso contrário} \end{cases}$$

onde o símbolo / corresponde a operação de divisão. Portanto o número cópias ($a(\alpha s_j \beta)$) produzidas para uma única derivação é dado por:

$$a(\alpha s_j \beta) = (\partial (\bar{a} (\alpha s_i \beta), \rho(d(s_i, s_j))) * A_{max})$$

onde ∂ é um operador nebuloso correspondente a uma co-norma (denominada de norma-s). As definições de norma-s e norma-t são apresentadas em [ZIME91]. O símbolo * corresponde ao operador produto.

Nas situações onde existam n derivações temos:

$$\begin{aligned} a(\alpha s_j \beta) = \ & max \ ((\partial (\bar{a} (\alpha s_i \beta), \rho(d(s_i, s_1)) * A_{max}), \\ & (\partial (\bar{a} (\alpha s_1 \beta), \rho(d(s_1, s_2)) * A_{max}), \\ & \ldots \\ & (\partial (\bar{a} (\alpha s_n \beta), \rho(d(s_n, s_j)) * A_{max})) \end{aligned}$$

$$a(\alpha s_j \beta) = (\max_{k=1}^{n} (\partial (\bar{a}(\alpha s_i \beta), \rho(d(s_i, s_j))) * A_{max}))$$

2.3.4 Tipos de Linguagens Nebulosas

A adoção de restrições no controle da ambigüidade das linguagem nebulosas mostradas na seção anterior permite a especificação de dois tipos de linguagem:

a) a Linguagem Nebulosa Simples (LNS): é especificada adotando as restrições d) e e); e,

b) a Linguagem Nebulosa Complexa (LNC): é especificada adotando as restrições a) e f).

Na seção anterior introduziu-se a quantidade das cadeias no processo de derivação. Assim, as quantidades disponíveis de $A(s_i)$, $A(s_k)$ de s_i, $s_k \in V^*$ são utilizadas na ativação da regra $\alpha s_k \beta \rightarrow \alpha s_j \beta$ para reescrita de s_i em s_j no processo de derivação $d(s_0,...s_i, s_j,...,s_t)$ da cadeia $s_t \in L(G_n)$. Para cada cadeia $s_t \in L(G_n)$ calcula-se o valor de ativação total $\mu_d(s_0,s_t)$. De acordo com o valor de ativação total $\mu_d(s_0,s_t)$ obtido, as seguintes regras de adaptação são aplicadas:

a) recompensa: o incremento da quantidade das cadeias utilizadas nas derivações ($A(s_i)$, $A(s_k)$) quando $d(s_0,...s_i, s_j,...,s_t)$ sucede em gerar s_t e $\mu_d(s_0,s_t) \rightarrow 1$;

b) punição: o decremento da quantidade das cadeias utilizadas nas derivações ($A(s_i)$,$A(s_k)$) quando $d(s_o,...s_i, s_j,...,s_t)$ sucede em gerar s_t e $\mu_d(s_0,s_t) \rightarrow 0.5$;

c) reconsideração: o decremento do casamento das cadeias ($\mu_c(s_i,s_k)$) utilizadas na derivações quando $d(s_o,...s_i, s_j,...,s_t)$ sucede em gerar s_t e $\mu_d(s_0,s_t) < 0.5$;

d) reforço: o incremento do casamento entre as cadeias ($\mu_c(s_i,s_k)$) utilizadas na derivações quando $d(s_o,...s_i, s_j,...,s_t)$ sucede em gerar s_t e $\mu_d(s_0,s_t) > 0.5$.

Desta forma, as regras acima permitem a implementação do aprendizado quantitativo (a, b) e qualitativo (c, d), e também podem ser usadas para reduzir a ambigüidade de uma linguagem nebulosa. Isto porque, a reconsideração e o reforço implicam na mudança da estrutura dos símbolos, e por outro lado, a punição e a recompensa estão associadas às mudanças no número disponível de cópias dos símbolos.

Como conseqüência, o aprendizado mostrado acima modifica a possibilidade $\rho(d(s_0,...,s_j,s_t))$ da ocorrência da cadeia de derivação $d(s_0,...,s_j,s_t)$. O valor de limite 0.5 utilizado nas regras está relacionado ao conceito da Teoria Nebulosa, onde 0.5 está relacionado ao mais alto grau de incerteza.

2.3.5 Definição das Derivações como Grafo

Uma outra forma para descrever as derivações em uma gramática é a estrutura denominada árvore de derivação. A forma utilizada para descrever as derivações neste trabalho é uma estrutura denominada *módulo*. Assim, *módulo* é um grafo nebuloso ([KAUF75]) que especifica as derivações segundo uma gramática nebulosa (G_n). A diferença fundamental entre árvore de derivação e um *módulo* consiste do fato de que os módulos são gerados segundo uma abordagem de baixo para cima, enquanto na outra, a geração é feita de cima para baixo (Figura 2.1). É importante destacar que nas formas de visualizar as derivações citadas acima, a topologia do grafo é criada de acordo com a gramática definida.

Um módulo é estruturado em três níveis: entrada, associativo e saída. No nível de entrada estão os nodos associados aos símbolos terminais. Neste nível, existe um ou mais nodos à esquerda, associados ao termo-chave ou germe, que caracterizam o módulo. No nível associativo estão os nodos representando as regras aplicadas, e no nível de saída estão os nodos representando as interpretações (Figura 2.1 (B)). A geração de um módulo é feita na seguinte seqüência: nível entrada, nível associativo e nível de saída.

Em virtude dos módulos serem criados a partir do nível de entrada, a especificação da gramática nebulosa apresenta alguns aspectos específicos. O primeiro aspecto consiste da necessidade da definição dos símbolos iniciais que podem ser elementos do dicionário dos símbolos terminais (V_T) ou o tamanho de um cadeia contendo símbolos terminais (V_T). A codificação na sintaxe é feita através de produções

na forma $\mathbf{V} \rightarrow \mathbf{vC}$, onde o **V** é um símbolo pertencente a S (conjunto de símbolos iniciais), sendo entretanto considerado o símbolo inicial na geração do módulo o símbolo terminal **v**. Em outras palavras, as produções dos símbolos de S relacionam os símbolos terminais (V_T) considerados como termo inicial. Na outra forma, o conjunto dos símbolos iniciais contém somente símbolos do conjunto de símbolos terminais e as produções são da forma $\mathbf{v} \rightarrow \mathbf{vC}$. As outras regras são codificadas para expressar as outras relações sintáticas existentes. Essas outras regras são ativadas a partir da verificação da presença de um símbolo inicial.

O início da geração de um módulo ocorre quando o símbolo inicial é detectado na sentença analisada. Assim, é criado o(s) nodo(s) mais a esquerda do grafo que é associado ao símbolo inicial. A presença do símbolo inicial permite que sejam disparadas as regras de produções que determinam os outros símbolos a serem reconhecidos no módulo. Evidentemente, os símbolos reconhecidos podem disparar regras, o encadeamento dos outros símbolos da sentença. Os encadeamentos são representados por nodos intermediários que conectam os símbolos utilizados nas regras. Os nodos de mais alto nível são os de saída e especificam as várias interpretações existentes. Neste contexto, as regras sintáticas são disparadas quando a parte da direita da regra acopla-se à cadeia analisada (Figura 2.1 (B)).

A seguir, é mostrada a geração dos módulos de acordo com duas especificações gramaticais: gramática simples e gramática complexa.

2.3.5.1 Módulos de uma Gramática Nebulosa Simples

As restrições de uma gramática nebulosa simples consiste do número limite de elementos de V_T nas cadeias analisadas e do número de símbolos no dicionário V_T.

Neste tipo de gramática, o número de símbolos em um símbolos inicial pode ser único ou composto (germe). O tamanho da cadeia e os símbolos iniciais são previamente definido. Assim, na análise de uma sentença é verificada a presença de algum símbolo inicial. Cria(m)-se o(s) nodo(s) mais a esquerda do grafo que é associado ao símbolo inicial. A presença do símbolo inicial permite que sejam disparadas regras de produções que determinam os outros símbolos que são reconhecidos no módulo. Desta forma, os símbolos que são reconhecidos podem ter regras associadas e, conseqüentemente, podem ter outros símbolos associados.

Cada regra ativada é representada no módulo por um nodo e pelos arcos que agregam os símbolos que possibilitaram a ativação da mesma. Os nodos são criados na medida que existam regras ativadas pelos símbolos existentes ou até que o número limite de símbolos encadeados seja alcançado. Os nodos de saída são criados para cada uma das interpretações existentes.

Dada uma gramática regular, onde $V_T = \{v, c\}$, $V_N = \{V, C\}$, $S = \{V\}$ e $P = \{V \rightarrow vC,\ C \rightarrow cC,\ C \rightarrow c\}$. As sentenças produzidas por esta gramática são da forma vc, vcc, vccc, assim, generalizando, temos $v\{c\}^i$. A geração do módulo na análise dessas sentenças é feita da seguinte forma: a ocorrência do símbolo **v** (associado ao símbolo inicial V) dá início à geração do módulo e provoca a reescrita de **vC**, que determina a ativação das regras associadas à **C**, e em seguida, provoca a verificação da existência de uma ou mais ocorrências de **c**. Um módulo gerado segundo esta gramática para a análise da sentença vcccc é mostrada abaixo.

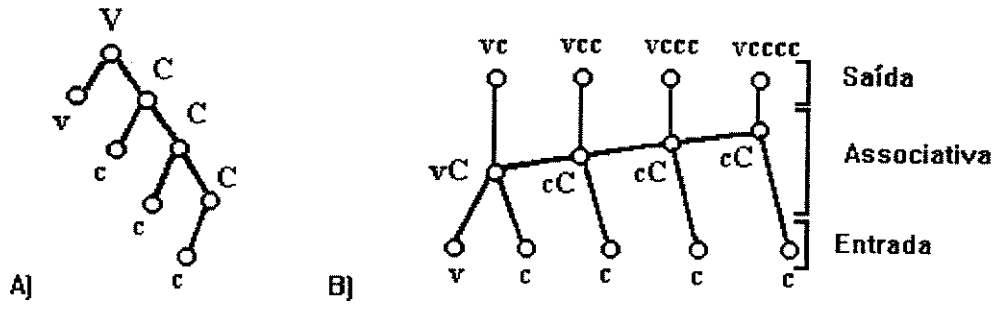

Figura 2.1: Árvore de derivação (A) e o correspondente Módulo (B).

No exemplo acima, se fosse estabelecido o número limite de quatro elementos na cadeia o módulo acima conteria somente três interpretações. A interpretação **vcccc** não seria uma interpretação válida por conter cinco elementos.

Uma sintaxe semelhante ao exemplo acima para a geração do módulo e o reconhecimento de uma frase do tipo "João comeu o bolo" é mostrada a seguir:

V → comeu C	C → João C	C → João
C → o C	C → o	C → bolo C
C → bolo.		

Na sintaxe acima, o símbolo inicial é o verbo "comeu", que dá inicio à criação do módulo e à verificação de um ou mais complementos na frase. Os complementos são verificados na frase de acordo com as regras sintáticas acima. Assim, são criados no módulo todas as interpretações mostradas a seguir e que são compreendidas como as interpretações da frase segundo a sintaxe definida.

comeu João	comeu o	comeu bolo
comeu João o	comeu João bolo	comeu o bolo
comeu João o bolo		

As interpretações nebulosas de uma frase são caracterizadas por não necessitarem conter todas as palavras das frases. Por esse motivo, a combinação "comeu João", que é reconhecida pela aplicação das regras V → comeu C e C → João é considerada uma interpretação possível. Algumas interpretações, tais como, "comeu João bolo" e "comeu bolo João" são sintaticamente corretas, sendo porém redundantes por conterem os mesmos símbolos.

As duas sintaxes definidas acima são codificadas tendo um símbolo inicial e um ou mais complementos. Como temos uma única classe sintática (C) nas produções associadas aos complementos, as derivações consistem, na verdade, das combinações dos complementos. Portanto, na geração dos módulos utilizando este tipo de sintaxe as derivações são combinatórias.

2.3.5.2 Módulos de uma Gramática Complexa

Uma outra análise da mesma frase "João comeu o bolo", porém utilizando regras sintáticas bem mais complexas, é apresentada abaixo. A geração do módulo da frase acima, segundo a sintaxe, é feita da seguinte forma: a ocorrência do símbolo "Comeu" dá início a geração do módulo e provoca a verificação da existência de regras associadas aos símbolos da cadeia SUJCOM; a verificação da presença do símbolo "João" é reconhecida por uma regra associada a cadeia SUJ; a verificação da presença do símbolo "o" é reconhecida por uma regra associada à cadeia COM, condicionando a existência de um símbolo associado a cadeia ARTOBJ; e finalmente, a presença dos símbolos "o" e "bolo" é feita pela regra associada à ART e OBJ, respectivamente. O módulo gerado é apresentado abaixo.

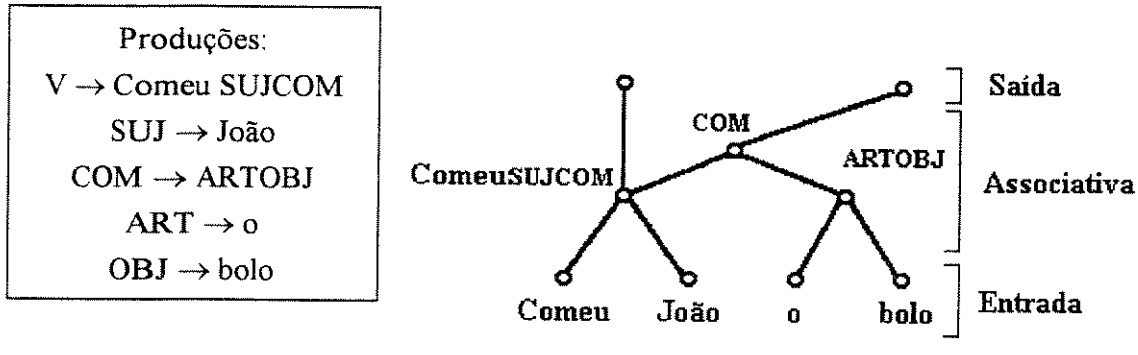

Figura 2.2: Módulo de uma Gramática Complexa.

No exemplo acima, o módulo criado para a análise da frase tem apenas dois nós de saída, desta forma, duas interpretações. Isto significa que a correta definição da sintaxe faz com que exista um número menor de interpretações específicas do contexto.

Os exemplos acima mostram que a complexidade ou a quantidade de interpretações nos módulos é maior quanto mais simples for a sintaxe adotada. Por outro lado, o módulo gerado com sintaxe complexa tem um número reduzido de interpretações.

2.3.6 Definição Formal do Módulo

Assim, um módulo gerado segundo uma gramática G_n é composto de um conjunto finito de nós conectados que satisfaçam as seguintes condições:

a) o grafo gerado possui três níveis: entrada, associativa e saída;

b) há um nodo correspondendo ao símbolo inicial ou nodos (germe) mais a esquerda na camada de entrada que inicia o processo de criação do módulo;

c) para cada nó da camada de entrada e associativa do grafo há um rótulo que é um símbolo em V ($V_N \cup V_T$). Os símbolos de V_T são associados aos nós da camada de entrada. Os símbolos de V_N são associados aos nós da camada associativa;

d) na camada de saída são criados tantos nodos de saída quanto forem as interpretações existentes no módulo.

2.4 Raciocínio Através de Conceitos

Segundo Patrícia Churchland, a linguagem ajuda a categorizar o mundo e a reduzir a complexidade da estrutura conceitual para uma escala possível de ser trabalhada. É muito comum encontrar num dicionário palavras que são associadas a vários conceitos. Assim, a um único símbolo podem existir vários conceitos associados. Esta economia apresentada pela linguagem, em representar vários conceitos em um único símbolo, é que torna possível a construção de conceitos mais complexos e de utilizá-los no raciocínio de mais alto nível ([DAMA92]).

O uso de conceitos para abstrair modelos da realidade é uma estratégia presente em nossa cultura que remonta da civilização grega ([SOWA84]), e é suportada por pesquisas oriundas da psicologia ([ORNS91]). Segundo Sowa ([SOWA84]), as unidades elementares, extraídas por circuitos neurais de baixo nível, são os "percepts" que são utilizados como informação básica na construção dos conceitos. Assim, a identificação de um objeto é feita pelo reconhecimento de um conjunto estável de relações que são estabelecidas entre os "percepts". Desta forma, os conceitos concretos são construídos,

através do aprendizado, como um conjunto de relações estáveis entre os "percepts" encontrados no mundo observado. Conceitos abstratos são construídos através do aprendizado e consistem de um conjunto de relações estáveis entre conceitos concretos e primitivos. Os conceitos primitivos são definidos como um conjunto de relações entre propriedades elementares suportadas por fatos encontrados no mundo observado. O nível hierarquicamente superior é o nível dos conceitos complexos, que é construído como relações estáveis entre os conceitos primitivos; e sucessivamente, temos as teorias que consistem das relações estáveis entre conceitos complexos.

Assim, uma abordagem conceitual pode ser definida da seguinte forma:

a) **C** um conjunto de conceitos;

b) **R** o conjunto das possíveis relações entre os conceitos;

c) **F** um conjunto de fatos sobre o universo **U**.

Uma teoria ou modelo **T** abordando **U** é um subconjunto de relações entre esses conceitos de **C** e suportado por fatos em **F**. Assim temos:

$$\mathbf{T} \subset \mathbf{R(C)} \Leftarrow \mathbf{F}$$

onde $\subset$ significa contido e $\Leftarrow$ denota "suportado por". Portanto, **T** é um conjunto de relações **R** entre os conceitos **C** que foram aprendidas ou provadas em **U**. Sendo também suportada por um conjunto de fatos de **F** extraídos de **U**.

A definição de um conceito de nível hierárquico mais alto é também uma teoria definida sobre os níveis precedentes. Sendo, portanto, possível definir os vários níveis das teorias:

$$(\mathbf{T_{i+1}} \subset \mathbf{R_{i+1}(C_{i+1})} \Leftarrow \ldots (\mathbf{T_1} \subset \mathbf{R_1(C_1)} \Leftarrow \mathbf{F}))$$

onde, $\mathbf{T_{i+1}}$ é definida por um conjunto de relações $\mathbf{R_{i+1}}$ entre os conceitos $\mathbf{C_{i+1}}$ suportado por fatos codificados no nível hierárquico inferior.

Uma base de dados em linguagem natural (BDLN) é considerada como um conjunto de fatos **F** sobre o universo do contexto **U** que se pretende analisar. Nesta BDLN, as palavras são composta de componentes elementares $\mathbf{C_1}$ (caracteres ASCII) e relacionados de acordo com $\mathbf{R_1}$ na composição dos conceitos primitivos $\mathbf{R_1(C_1)}$. Os conceitos primitivos obtidos na BDLN são selecionados de acordo com o propósito da análise a ser executada. Portanto, essas palavras são consideradas como os símbolos que especificam os conceitos primitivos $\mathbf{R_1(C_1)}$ no contexto analisado, suportando uma teoria local $\mathbf{T_1}$ de acordo com os fatos de **F**.

O próximo nível hierárquico de conhecimento a ser obtido é o de conceitos complexos $\mathbf{R_2(C_2)}$ suportados por **F**. Isto implica em obter as relações $\mathbf{R_2}$ entre conceitos primitivos $\mathbf{C_2}$ que correspondem ao nível conceitual inferior $(\mathbf{R_1(C_1)})$ suportados por **F**. Numa linguagem **L**, as relações possíveis entre as palavras são definidas através de uma gramática **G**. Assim, as relações $\mathbf{R_2}$ correspondem as regras

gramaticais no nível de frases. As relações $\mathbf{R_2}$ são definidas previamente e codificam as relações sintáticas entre os conceitos primitivos ($\mathbf{R_1(C_1)}$) existentes em **F**. Portanto, os conceitos complexos existentes em **F** são aquelas possíveis relações de palavras de acordo com $\mathbf{R_2}$. Os conceitos complexos suportado por **F** são chamados de teoria local $\mathbf{T_2}$ sobre **U**.

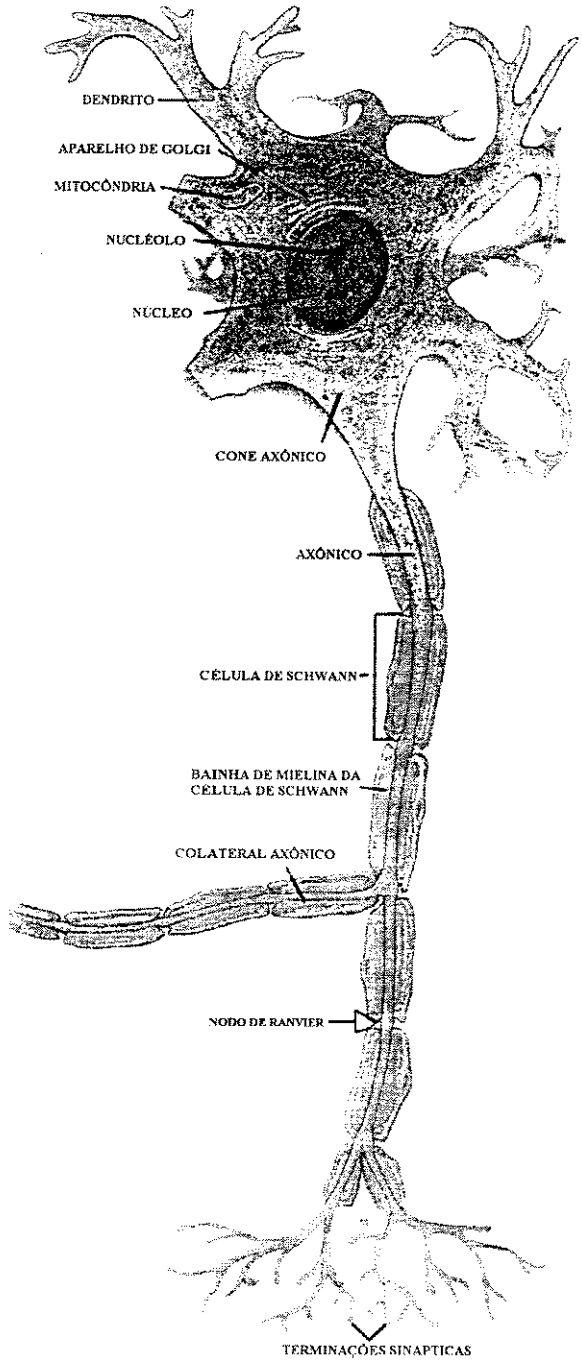

Figura 2.3 : Estrutura do neurônio.

Finalmente, é possível obter uma teoria $\mathbf{T_3}$ sobre **U** suportada por toda a base de dados **F**. Neste nível, utiliza-se os conceitos complexos $\mathbf{C_3}$ que consiste do nível conceitual hierarquicamente inferior $\mathbf{R_2(C_2)}$ e que são igualmente suportados em todas as teorias locais. Assim, $\mathbf{R_3}$ é uma gramática definida para codificar as relações entre os elementos de $\mathbf{C_3}$, suportadas por **F**. O nível de teoria $\mathbf{T_3}$ corresponde ao nível dos textos ou diálogos presentes em **F**.

2.5 Aspectos Conceituais do Conexionismo

O neurônio é a unidade básica de processamento no cérebro. Sua descrição básica é mostrada na Figura 2.3, e consiste: de ramificações denominadas de dendritos, parte onde são recebidas as informações; de um corpo celular, parte onde encontra-se o núcleo e as informações são agregadas; e do axônio onde é propagada as informações de saída.

O modelo de neurônio (Figura 2.4) utilizado em grande parte das arquiteturas conexionistas segue o modelo de neurônio definido por McCulloch e Pitts. Ele consiste de **n** entradas, sendo que cada entrada **i** possui um peso $\mathbf{w_i}$ associado. No núcleo é feita a agregação (somatória) dos valores de ativação das entradas $\mathbf{x_i}$ ponderadas pelo peso, sendo então subtraído o limiar de ativação θ. A saída é obtida aplicando-se uma função **f** ao valor resultante das operações efetuadas no núcleo.

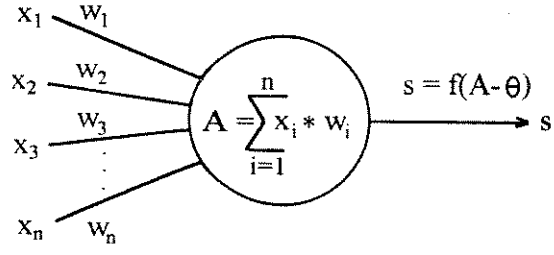

Figura 2.4: Modelo clássico de neurônio usado em Redes Neurais.

Este modelo clássico de neurônio, devido à sua simplicidade, tem sido utilizado basicamente como um modelo de processamento numérico. Por outro lado, nas últimas décadas as neurociências têm produzido novos conhecimentos dos mecanismos do cérebro, e grande parte deles estão relacionados com os mecanismos de funcionamento dos neurônios. Entretanto, tem sido verificado que muito pouco desses novos conhecimentos são utilizados nas arquiteturas conexionistas. O que freqüentemente ocorre, é uma maior influência de uma modelagem matemática nessas arquiteturas, aumentando sua disposição para o processamento numérico de informações.

2.5.1 O Processamento no Neurônio

A chegada do potencial de ação no axônio da célula pré-sináptica provoca o incremento da concentração intracelular de Cálcio (Ca) (Figura 2.5). Como conseqüência, é ativada a conversão de ATP (Adenosina Tri-Fosfato) em ADP (Adenosina Di-Fosfato), aumentando a quantidade de energia metabólica. Esta energia é usada por proteínas contrácteis para mover as vesículas de transmissores através da membrana celular. O transmissor é uma molécula usada para transmitir mensagens do potencial de ação das células pré-sinápticas para as pós-sinápticas.

A vesícula transportada funde-se com a membrana e libera o transmissor (t) no processo denominado de exocitose. O transmissor difunde-se até a célula pós-sináptica e liga-se a uma molécula chamada receptor (r). Receptor é uma molécula que tem alta afinidade com o transmissor t. O acoplamento entre transmissor e receptor ativa algumas moléculas chamadas de atuadores ou controladores. Assim temos:

$$t \wedge r >> c$$

sendo:

^ a operação de acoplamento entre transmissor e receptor;

>> é a operação de ativação do controlador.

Os atuadores exercem ação sobre as células pré, pós-sinápticas e também sobre as células vizinhas. A ação exercida pelo atuador depende da sua estrutura e função, podendo atuar sobre:

a) canal iônico: o acoplamento t ^ r modifica a permeabilidade pós-sináptica e isto promove a modificação do potencial da membrana (PM), provocando a hiperpolarização e despolarização;

b) enzimas controlando algumas cadeias metabólicas: o acoplamento t ^ r muda a quantidade de energia disponível na membrana. Em geral, esta energia é usada para modificar o limiar da membrana;

c) molécula de controle ou reguladora: o acoplamento t ^ r dispara uma das seguintes ações:

- modulação do acoplamento t ^ r ;
- especificação da leitura do ácido desoxirribonucléico (ADN) no núcleo celular;
- ativação da leitura do ADN no núcleo celular;
- especificação da síntese de proteínas de moléculas definidas.

As intensidades sinápticas produzidas pelo t ^ r >> c dos diferentes neurônios pré-sinápticos são agregadas no axônio do neurônio pós-sináptico. Freqüentemente, nos modelos de neurônios artificiais esta agregação é feita utilizando o operador soma. O valor da saída no axônio é produzido pela aplicação de uma função linear g sobre o valor

agregado das diferentes atividades sinápticas. Este valor produzido atua em seguida na liberação de uma quantidade de transmissor.

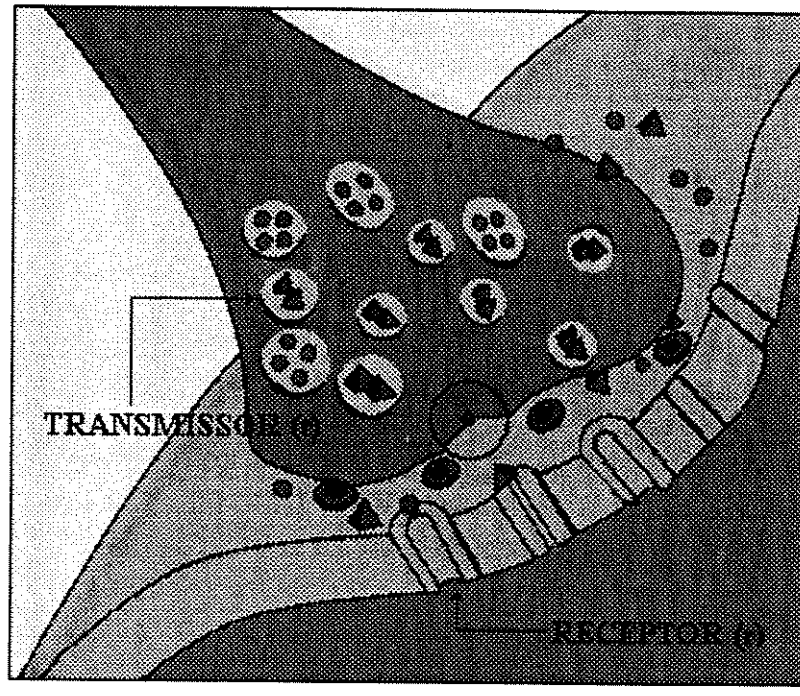

Figura 2.5: A sinapse.

2.5.2 Especificação Formal da Linguagem T^R >> C

Seja L um conjunto de símbolos composto de todas as letras minúsculas e maiúsculas, números e os caracteres # e ?.

$$L=\{a,b,...,y,z,A,B,...Z,1,2,...,9,\#,?\}$$

Os transmissores, controladores e receptores são cadeias construídas de acordo com uma gramática, sendo composta de elementos de L.

Sejam os seguintes subconjuntos de L:

$$L_1= \{a,b,...,l,\#,?\},$$

$$L_2 =\{m,n,...,z,\#,?\},$$

$$L_3= \{A,B,...,Z,\#,?\},$$

$$L_4= \{0,1,...,9,\#,?\}.$$

As cadeias dos transmissores T são geradas pela gramática ϕ_1 e são compostas por caracteres contidos em L_1 com comprimento N.

$$\phi_1: L_1^N \rightarrow T$$

As cadeias dos receptores R são geradas de acordo com uma gramática ϕ_2 e são compostas por caracteres contidos em L_3 com comprimento O.

$$\phi_2: L_3^O \rightarrow R$$

As cadeias dos controladores C são geradas de acordo com uma gramática ϕ_3 e são compostas pelas subcadeias C_r e C_l. As subcadeias C_r são de comprimento P e compostas pelos caracteres contidos em L_2. As subcadeias C_l são de tamanho S e compostas por caracteres contidos em $L5 = L_1 \cup L_3 \cup L_4$ ou $L5 = L_1 \cup L_3$. Assim, os controladores são gerados como:

$$\phi_3: L_2^P \times L_5^S \rightarrow C$$

A cadeia C_r é a subcadeia onde ocorre o acoplamento, e a subcadeia C_l é onde está especificada a ação do controlador.

O conjunto das cadeias utilizadas para descrever os transmissores, controladores e receptores é denotada por LF. Onde temos:

$$LF = T \cup R \cup C$$

2.5.2.1 Afinidade entre elementos

Os alfabetos associados às classes dos transmissores, receptores e controladores devem ser especificados de tal forma que possam representar os acoplamentos que ocorrem entre as classes. Seja um dicionário L_i, cujos elementos s_i estão presentes nas cadeias associadas a uma classe *i* que irá ser acoplada a uma outra classe *j* representada por cadeias geradas com elementos s_j do dicionário L_j. O acoplamento entre elementos de L_i e L_j ocorre em função da afinidade μ entre eles.

$$\mu: L_i \times Lj \rightarrow [0,1]$$

Portanto, a afinidade do elemento $s_i \in L_i$ e $s_j \in Lj$ é denotada por $\mu(s_i,s_j)$. Se $\mu(s_i,s_j) > 0.5$, então s_i e s_j são chamados símbolos complementares. Na definição da LF na seção anterior, os símbolos $s_j \in L_3$ são especificados através de letras maiúscula e $s_i \in L_1 \cup L_2$ em letras minúsculas. A afinidade entre um s_j e o seu correspondente caractere em letra minúscula s_i é:

$$\mu(s_i,s_j) = 1$$

$$\mu(\# \text{ ou } ?,s_j) = 1$$

$$\mu(s_i, \# \text{ ou } ?) = 1$$

Os caracteres # e ?, quando presentes, se acoplam a qualquer outro caractere igual ou diferente deles.

2.5.2.2 Afinidade entre cadeias

Seja α uma cadeia associada à classe *i*, composta por elementos de L_i, e seja β uma cadeia associada à classe *j*, composta por elementos de L_j. Os elementos da classe *i* se acoplam aos da classe *j*.

$$\alpha = s_1,\ldots,s_m \qquad \beta = s_1,\ldots,s_n$$

Se α e β não possuem símbolos complementares, então não ocorre o acoplamento. Se todos os símbolos de α e β são complementares, então o acoplamento é dito completo (Figura 2.6.A). Caso contrário, é dito parcial (Figura 2.6.B).

O cálculo da afinidade $\mu_c(\alpha, \beta)$ entre as cadeias α e β de comprimento respectivamente m e n, consiste em agregarmos as afinidades entre os caracteres complementares $s_t \in \alpha$ e $s_r \in \beta$. Portanto, o cálculo da afinidade é dado por:

$$\mu(\alpha,\beta) = \left(\sum_{t=1}^{m}\sum_{r=1}^{n}\mu(s_t, s_r)\right) / min(m, n)$$

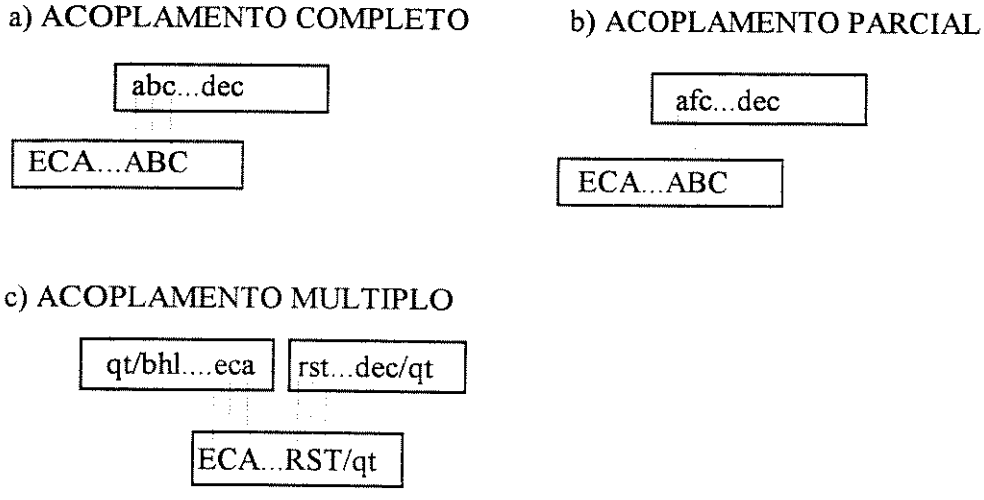

Figura 2.6: Concatenação das cadeias.

2.5.3 Processamento Químico

O processo sináptico envolve um encadeamento químico do tipo

$$T \wedge R >> C \rightarrow \text{AÇÕES}$$

representando:

a) um conjunto T de transmissores: cada elemento $t \in T$, especificado segundo a LF definida na seção 2.5.2, pode ser atribuído em diferentes quantidades aos terminais do axônio pré-sináptico;

b) um conjunto R de receptores: cada elemento $r \in R$, especificado segundo a LF definida na seção 2.5.2, pode ser atribuído em diferentes quantidades às células pós-sináptica;

c) um conjunto C de controladores: cada elemento $c \in C$, especificado segundo a LF definida na seção 2.5.2, pode ser ativado pelo acoplamento T e R.

As ações são modificações da fisiologia da célula pré e pós-sináptica induzidas pela ativação da sinapse.

Para exemplificar, **ABCDEF** pode ser uma cadeia associado a um receptor, **abc** pode ser uma cadeia associada a um transmissor e **defMNR** uma cadeia associada a um controlador.

Após especificar os símbolos associados aos transmissores, receptores e controladores, especifica-se também as suas quantidades nos neurônios. Assim, temos:

a) A(t) a quantidade do transmissor t armazenado no neurônio pré-sinápticos, e

b) A(r) a quantidade do receptor r armazenado no neurônio pós-sinápticos.

Então:

c) a função Γ codificada no neurônio pré-sináptico define a quantia m_i de transmissor liberada quando a atividade é v_i no neurônio pré-sináptico,

$$m_i = \Gamma(v_i, A(t))$$

Onde Γ em geral é uma norma-t do tipo mínimo ou produto algébrico.

d) Se existe uma compatibilidade $\wedge$ entre T e R, então o acoplamento ocorre e a quantidade de resultante do controlador é dada por:

$$a(c) = m_i \otimes A(r) \oslash \mu(t, r)$$

onde, $\otimes$ e $\oslash$ são norma-t do tipo mínimo ou produto algébrico.

A regra de compatibilidade ^ das cadeias de T e R é definida como o seguinte mapeamento nebuloso:

$$\wedge: T \times R \rightarrow [0,1]$$

$$\mu(t_i,r_i) \rightarrow 0, \text{ se } t_i \text{ não concatenar com } r_i;$$
$$\mu(t_i,r_i) \rightarrow 1, \text{ caso contrário.}$$

Portanto, $\mu(t_i,r_i)$ mede a possibilidade de $t_i \in T$ concatenar-se com $r_i \in R$. Nessas condições, o valor w_i da sinapse torna-se:

$$w_i = f(\mu(t_i,r_i), A(t_i), A(r_i))$$

e a quantia de controlador a(c) produzida no neurônio pós-sináptico é:

$$a(c) = \partial(v_i, w_i).$$

Em geral, temos:

$$a(c) = v_i * w_i$$

onde, * denota o produto algébrico e ∂ é uma norma-t.

As operações ^ e >> são usadas no processo de encadeamento (Figura 2.7). Por exemplo, o transmissor **eca** acopla-se com a subcadeia **ECA** do receptor **ECA...RTS.** Pode então ocorrer a mudança do sítio de acoplamento de **RTS** para **DEC** pela concatenação do controlador **rts...DEC** para o local **RTS**. A concatenação pode também ser vista como a substituição de **RTS** pelo controlador **rts...DEC.**

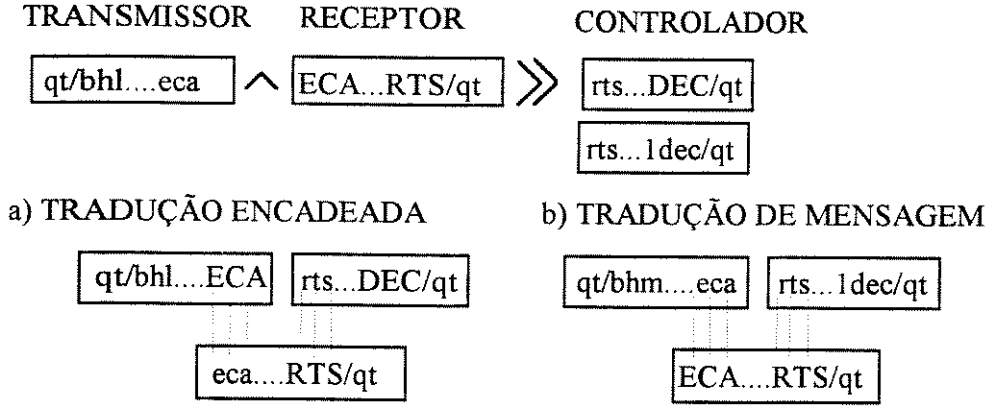

Figura 2.7: Encadeamento t^r >> c

A implementação prática deste encadeamento é apresentada na Figura 2.7 e pode ser descrita por:

a) ^ é um encadeamento entre L5 e L2 da seguinte forma:

$$\wedge: L5 \times L2 \rightarrow \{0,1\}$$

onde, se um caractere minúsculo $d_r \in R$ e $d_t \in T$ é o caractere maiúsculo correspondente, então $\mu(d_r,d_t)=1$. Caso contrário $\mu(d_r,d_t)=0$. Exemplificando, temos $\mu(a,A) = 1$ e $\mu(a,C)=0$.

b) ^ é um encadeamento entre T e R, tal que

$$\wedge: T \times R \rightarrow [0,1]$$

sendo calculado por:

$$\mu(t_i,r_i)=(\sum_{t=1}^{m}\sum_{r=1}^{n}\mu(d_r,d_t))/\min(m,n)$$

Sintetizando, a compatibilidade entre t_i e r_i é dada pela soma das compatibilidades entre os elementos da cadeia, dividido pelo tamanho da menor cadeia.

A operação >>, denominada de disparo ou tradução, é um caso especial de concatenação entre o produto do acoplamento t^r e $c \in C$, onde $c=c_l+c_r$. O resultado desta tradução depende do tipo da subcadeia c_r:

$$c_l \text{ e } c_r \in L5 = L1 \cup L2 \cup L4$$

$$t\wedge r >> c = c_l + c_r$$

Neste caso, c_r é produzido de C como uma mensagem para ativar algum outro processo.

Nestas condições, c_r é usado para especificar a saída do acoplamento t^r. Por exemplo, c_r é usado para controlar a síntese protéica do dicionário S; ou para especificar o gene de G a ser lido; ou para controlar a quantia de energia disponível para diferentes processos celulares, etc. Este tipo de operação é denominada de tradução de mensagem. A mensagem é composta por uma subcadeia do endereço e o tipo de ação executada pelo controlador.

$$t\wedge r >> c = c_l$$

onde c_l é uma cadeia de T, R, ou C.

Nestas condições, c_r é usado para condicionar os acoplamentos t^r. Esta mesma proteína c_r pode exibir diferentes pontos de atividades b_k, e o atual b_k é especificado por condições locais ou de contexto. Isto é implementado aqui como concatenação encadeada.

2.6 Redes Neurais Evolutivas (RNE)

As redes neurais evolutivas (RNE) ([ROCH92c]) são estruturadas em três níveis hierárquicos:

a) nível de entrada: formado por somente uma camada de neurônios;

b) nível intermediário ou associativo: formado por uma ou mais de uma camada de neurônios

c) nível de saída: formado por somente uma camada de neurônios;

Os neurônios da camada de entrada da rede RNE são criados para efetuarem sinapses com os neurônios da camada de saída ou da camada intermediária. Os neurônios das camadas intermediárias conectam-se com os neurônios da própria camada intermediária e com neurônios da camada de saída. Os neurônios das camadas de saída podem não se conectar a outro neurônio ou conectar-se a neurônios de entrada de alguma outra rede. As redes neurais RNE podem ser conectadas, isto ocorre quando neurônios das camadas de saídas de alguma rede neural RNE_i (rede neural de nível hierárquico i) conectam-se a neurônios de entrada de alguma outra rede neural RNE_{i+1} (rede neural de nível hierárquico i+1). O nível de hierarquia que é alcançado pela conexão entre as redes RNE e depende do nível de hierarquia da informação processada.

As redes neurais RNE contêm sub-redes denominadas de módulos. Os módulos têm a seguinte estrutura:

a) possuem um ou mais neurônios na camada de entrada podendo ser dividas em duas partes: germe ou símbolo inicial e halo ou complemento;

b) possuem uma ou mais camadas no nível intermediário;

c) possuem um ou mais neurônios na camada de saída.

O módulo é construído tendo na camada de entrada um neurônio ou o grupo de neurônios mais a esquerda, denominado de símbolo inicial ou germe, respectivamente. O germe e o símbolo inicial correspondem aos símbolos iniciais da gramática e, conseqüentemente, iniciam o processo de criação ou ativação do módulo. Os neurônios da camada de entrada acoplam-se aos neurônios da camada intermediária de acordo com as derivações existentes. Os neurônios da camada intermediária acoplam-se aos neurônios da própria camada e aos neurônios da camada de saída. Os neurônios da camada de saída caracterizam as várias interpretações incorporadas no módulo a partir do símbolo inicial ou do germe.

2.6.1 Geração da Rede Neural Evolutiva (RNE)

As RNE são geradas para incorporar, segundo uma gramática nebulosa G_n, as informações contidas em um conjunto de treinamento de um domínio. Portanto, o processo de geração das RNE ocorre obedecendo, fundamentalmente, a uma gramática G_n que define a topologia dos módulos da rede. A complexidade da sintaxe adotada é dependente do conhecimento disponível do contexto trabalhado. Outro aspecto importante considerado é o nível do conhecimento conceitual (seção 2.4) a ser codificado na RNE. Assim, de acordo com a sintaxe, com a complexidade do conhecimento disponível e o nível do conhecimento conceitual são gerados os seguintes tipos de RNE: as RNE de Conceitos Primitivos; e as RNE Conceitos Complexos.

A geração de uma rede RNE de Conceitos Primitivos é feita para codificar os conceitos primitivos existentes no conjunto de treinamento. A gramática especificada é uma gramática G_n simples, na qual define-se o número de símbolos presentes no germe e halos.

A geração da RNE complexa é feita para codificar o conhecimento de mais alto nível (Conceitos Complexos, Teorias, etc) existente no conjunto de treinamento. A gramática é codificada de acordo com o conhecimento do domínio do problema ao qual corresponde o conjunto de treinamento. Assim, uma gramática G_n simples é especificada quando o conhecimento do conjunto de treinamento é pequeno. Uma gramática G_n complexa é especificada quando grau de conhecimento do conjunto de treinamento é razoável.

A geração e todos os processos de propagação das informações nas RNE são efetuados partindo-se primeiramente do nível de entrada, seguido do nível associativo e finalmente, nível de saída. Nas camadas do nível associativo são criados, primeiramente, os neurônios das camadas inferiores. Esta abordagem segue o paradigma conexionista que tem como uma das principais, características a geração da estrutura da rede e propagação de informações a partir do nível de baixo (nível de entrada) para cima (nível de saída).

2.6.2 Geração dos Módulos da RNE

2.6.2.1 Geração dos Módulos nas RNE de Conceitos Primitivos

Os módulos das redes neurais de Conceitos Primitivos são caracterizados por terem os neurônios da camada de entrada agrupados em partes:

a) a parte denominada *germe* é correspondente ao conjunto de neurônios mais a esquerda e que são agregados por um neurônio da camada intermediária.

b) a parte denominada *halo* corresponde aos neurônios que complementam o germe e que são agregados por um neurônio da camada intermediária.

Os neurônios da camada intermediária são agregados por neurônios da camada de saída.

Na geração de um módulo na RNE de Conceitos Primitivos a especificação da gramática simples consiste, em outras palavras, da definição do número máximo de neurônios no germe e nos halos a serem criados:

a) o símbolo inicial (germe) é uma cadeia composta de no mínimo **k** e no máximo **n** símbolos do tipo $v_1\ v_2\ ...v_k...v_n$, e que reconhece os primeiros **n** caracteres da cadeia de entrada; e

b) a presença do símbolo inicial dá início ao reconhecimento de até **m** símbolos complementares $c_1,c_2,...c_m$ (halo). Assim, o tamanho máximo **t** da cadeia reconhecida é dado por: $t = n+m$.

O início da geração é feito a partir de um conjunto de treinamento. Cada sentença do conjunto de treinamento é considerada como uma cadeia de transmissores, isto porque cada um dos símbolos que compõe a sentença corresponde a um transmissor.

Assim, cada vez que uma sentença do conjunto de treinamento é lida, percorre-se todos os módulos existentes na rede e verifica-se a ocorrência do acoplamento entre os transmissores da parte inicial da sentença e os receptores da parte denominada germe da camada de entrada de algum módulo. Dependendo do resultado do acoplamento, as ações são aplicadas de acordo com as seguintes regras:

a) se não ocorrer o acoplamento com nenhum dos módulos existentes, é criado um novo módulo. No módulo, são criados até **t** neurônios (número máximo de símbolos na cadeia) na camada de entrada, e cada neurônio criado contém receptores para os transmissores dos símbolos se acoplarem. Por sua vez, os neurônios de entrada produzem transmissores para acoplarem-se a um neurônio do nível de saída. Desta forma, para toda sentença nova encontrada no conjunto de treinamento, um módulo novo é criado na rede.

b) Se ocorrer o acoplamento na parte denominada germe, e ocorrer também o acoplamento entre a parte restante da sentença (parte denominada halo) e algum halo do módulo, então o valor das conexões sinápticas dos neurônios ativados no módulo são incrementados.

c) Se ocorrer o acoplamento na parte denominada germe, mas não ocorrer na parte restante da sentença (halo), então no mesmo módulo da rede é incorporado um novo conjunto de neurônios com receptores para acoplarem-se com os transmissores dos símbolos do halo da sentença. Os neurônios do halo produzem transmissores que se acoplam a um neurônio criado na camada intermediária. Os neurônios da camada intermediária que agregam o germe e o halo produzem transmissores que acoplam-se a um neurônio criado na camada de saída. Os valores das conexões sinápticas do germe são também incrementados.

2.6.2.2 Geração dos Módulos na RNE de Conceitos Complexos

O primeiro passo na geração das RNE de Conceitos Complexos utilizando uma gramática G_n simples ou complexa consiste na definição do dicionário de símbolos D. No contexto do sistema JARGÃO, o dicionário inicial é o de Conceitos Primitivos e que compõe o dicionário de símbolos D_1. Os dicionários subseqüentes têm seus módulos da rede, gerados de acordo com os símbolos contidos no dicionário anterior, assim sendo: D_2 contém as frases geradas de acordo com D_1, D_3 contém as estruturas criadas de acordo com D_2. Portanto, o dicionário D_{i+1} é criado de acordo com os símbolos contidos no dicionário D_i do nível anterior. O número de níveis é dependente do nível hierárquico do conhecimento.

A definição da linguagem $L_i(G_n)$, suportando uma gramática G_n, é especificada visando a codificação do conhecimento na RNE_i. A complexidade da sintaxe G_n é que vai definir a complexidade da topologia da RNE_i e também da estrutura do conhecimento nela contido.

A especificação dos símbolos da linguagem $L_i(G_n)$ deve ser feita de acordo com as definições feitas na seção 2.5.2. Após a definição, deve-se efetuar a atribuição dos símbolos da linguagem $L_i(G_n)$ para os elementos do dicionário D_{i-1}, que correspondem à atribuição de transmissores, receptores e controladores aos símbolos de D_{i-1}. Na verdade, consiste em atribuir os símbolos da linguagem $L_i(G_n)$ correspondentes aos transmissores, receptores e controladores para os neurônios da camada de saída da rede RNE_{i-1}, isto porque os elementos de D_{i-1} correspondem aos módulos existentes na RNE_{i-1} (Figura 2.8). Nesta fase de atribuição são também definidos os símbolos de D_{i-1} a serem considerados como termo-chave.

Portanto, o processo de geração das RNE_i é feita da seguinte forma:

$$(D_{i-1} \Leftarrow L_i(G_n)) \rightarrow RNE_i$$

onde, D_{i-1} é o dicionário e $L_i(G_n)$ são os símbolos de uma gramática G_n simples ou complexa. O símbolo $\Leftarrow$ representa a atribuição dos símbolos de $L_i(G_n)$ para os elementos do dicionário D_{i-1} e o símbolo $\rightarrow$ representa o processo de geração e treinamento da rede.

A geração dos módulos é feita tendo-se um conjunto de treinamento. Cada elemento presente no conjunto de treinamento ativará neurônios da camada de entrada (módulos do dicionário D_{i-1}), que por sua vez liberam transmissores que conectam-se aos receptores dos neurônios do nível de entrada da rede RNE_i. Se o módulo do dicionário ativado (módulos do dicionário D_{i-1}) é definido como termo-chave e não existem neurônios da camada de entrada com receptor correspondente, cria-se no nível de entrada da rede RNE_i um novo módulo tendo um neurônio com o receptor correspondente. Nessas condições, o sistema gerará tantos módulos na rede RNE_i quanto forem os termos-chaves definidos pelo usuário no dicionário D_{i-1}.

No novo módulo, os neurônios da camada de entrada, com receptores aos quais os transmissores associados aos complementos se acoplarão, são criados somente se existir um encadeamento entre:

a) o transmissor produzido pelo termo-chave e o receptor de um complemento; ou

b) o controlador produzido por um neurônio da camada associativa, resultado de outro acoplamento, e o receptor de um complemento.

Os neurônios da camada associativa são criados para codificar os acoplamentos existentes. Portanto, ocorrendo o encadeamento, os neurônios (termo-chave e complemento) de entrada ativados produzem transmissores que irão acoplar-se a um neurônio criado na camada associativa. Em seguida, cria-se o neurônio de saída ao qual o neurônio ativado da camada associativa se acopla. Deve-se salientar que os neurônios da camada associativa representam o encadeamento sintático e os neurônios de saída representam os módulos de interpretações da RNE na sintaxe codificada.

Os encadeamentos sintáticos reproduzem a ocorrência das sinapses entre os neurônios. Como conseqüência das sinapses, ocorrem a alteração da quantidade t_i de transmissores, c_k de controladores e r_j de receptores. A representação das sinapses nas redes é feita através dos arcos conectando os neurônios.

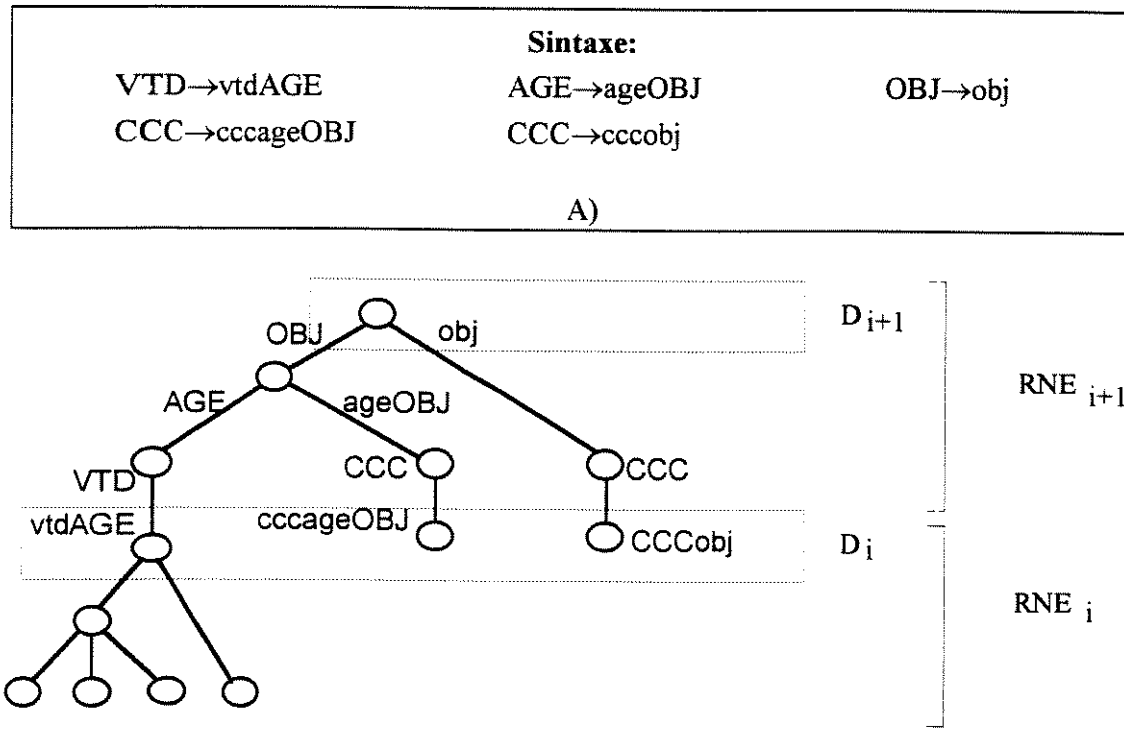

Figura 2.8: Definição da sintaxe (A). Módulo de uma RNE de acordo com a sintaxe (B).

Portanto, generalizando, a estrutura dos módulos é a seguinte:

a) são criados tantos módulos nas redes quanto forem o número de termos-chaves definidos ou germes existentes;

b) um ou mais neurônios a esquerda da camada de entrada do módulo é denominado de símbolo inicial ou germe. Esses neurônios caracterizam o módulo;

c) os outros neurônios de entrada são complementos e eles produzem receptores para diferentes categorias sintáticas aceitas pelos termos-chaves e seus próprios complementos. Nesse caso, são criados tantos neurônios de entradas quantas forem as classes sintáticas requeridas pelos termos-chaves e seus complementos;

d) são criados neurônios intermediários para agregarem os encadeamentos, que foram criados a partir do neurônio associado ao termo-chave com os outros neurônios da camada de entrada;

e) são criados neurônios de saída para agregar os neurônios dos níveis e camadas inferiores, e cada neurônio de saída está associado a um módulo de interpretação;

f) os arcos entre os neurônios representam as sinapses.

2.6.3 Definição Formal da RNE

Após a geração dos módulos a estrutura da RNE criada é descrita por:

$$RNE = \{ E, N, C, W, S \}$$

onde:

a) E é o conjunto dos neurônios de entradas da RNE associado a elementos do dicionário;

b) N é o conjunto dos neurônios da camada associativa gerada de acordo com a sintaxe G e que representa os encadeamentos sintáticos;

c) C é o conjunto das sinapses entre os neurônios;

d) S é o conjunto dos neurônios de saída;

e) W é o conjunto contendo os valores das conectividades w entre os neurônios. Sejam n_i e n_j os neurônios conectados pela sinapse c_i, então o valor da conectividade entre os dois neurônios é representada por w_{ij}.

2.7 Aprendizado Evolutivo

O aprendizado evolutivo ([ROCH92a et alli]) é uma técnica para desenvolver e treinar redes neurais evolutivas. Consiste em ajustar a entropia da RNE de acordo com a variabilidade do ambiente. Os passos deste aprendizado são os seguintes:

a) gênese: este passo consiste na geração da variabilidade estrutural das redes, criando quantos módulos forem necessário para representar as diferentes mensagens trocadas entre os módulos e acomodar as pequenas variações das mensagens;

b) adaptação: a exposição das redes a informações do meio externo causa a modificação das associações sinápticas dos módulos;

c) seleção: selecionar as redes que tiveram maior ativação, eliminando as de menor ativação. Pode-se aplicar dois mecanismos de seleção:

- a poda automática consiste da eliminação dos módulos de redes com cujo valor de entropia está abaixo de um limiar. O limiar pode ser fornecido ou obtido em função da variabilidade das redes ([ROCH92b et alli]).
- A poda seletiva consiste em permitir, se desejado, a eliminação ou alteração dos módulos obtidos no processo de treinamento.

2.7.1 Adaptação dos Módulos

O processo de adaptação consiste em incrementar todas as conexões sinápticas de um módulo, quando a ele acoplam-se dados presentes no conjunto de treinamento. O valor do aumento na conexão sináptica é a_k, cujo valor é igual à ativação do neurônio pós-sináptico n_k, e é obtido pelo aumento proporcional à a_k na quantidade de transmissor no neurônio pré-sináptico n_i, de receptor e de controlador no neurônio pós-sináptico n_k. Portanto, se ocorrem M acoplamentos entre n_i e n_k, temos o seguinte valor para transmissor (t_i), receptor (r_j) e controlador (c_k):

$$t_i(M)=t_i(M-1)+a_k$$

$$r_j(M)=r_j(M-1)+a_k$$

$$c_k(M)=c_k(M-1)+a_k$$

Desta forma, o valor das conexões sinápticas codifica a freqüência da ocorrência de uma instância presente no conjunto de treinamento.

2.7.2 Poda Automática

Após os módulos de uma RNE terem sido gerados, têm-se nas conexões as freqüências de cada conceito no conjunto de treinamento. O valor da freqüência pode ser utilizado para o cálculo da entropia estrutural da rede (RNE). Seja w_i o valor normalizado da conexão de maior valor do módulo i da rede. Portanto, na formulação de entropia, de acordo com Shanon, temos:

$$h(\mathrm{RNE}) = - \sum_{i=1}^{n} w_i * \log w_i$$

onde, n é o número de módulos na rede.

O valor da entropia h(RNE) da rede permite verificar a capacidade de informação obtida no conjunto de treinamento. A seguir, propõe-se um método visando maximizar a capacidade de informação contida nas redes. Este método consiste na implementação do conceito de entropia nebulosa, que é mostrado a seguir:

$$\max h(\mathrm{RNE}) = - \sum_{i=1}^{n} P_i * \log P_i$$

onde, n é o número de módulos na rede e Pi é a probabilidade do módulo i na RNE. A

probabilidade *Pi* do módulo i é calculada através da divisão da freqüência do módulo i pela somatória da freqüência dos módulos da RNE, portanto:

$$\sum_{i=1}^{n} P_i = 1$$

Os módulos da rede de baixíssima freqüência e de muita freqüência são de baixo valor de entropia, portanto, representam informações de pouca importância. A forma de eliminar os módulos de baixo valor de entropia é feita através da adoção do limiar α.

O valor do limiar α é obtido da seguinte forma:

$$\alpha = \beta * h(RNE) / n$$

onde, β é um valor no intervalo [0,1] que é especificado pelo usuário, h(RNE) é o valor da entropia estrutural da rede e n é o número de módulos na RNE. Após calcular o valor de α, seu valor é utilizado para eliminar todos os módulos com entropia menor, desta forma temos:

$$h(RNE) \geq \alpha$$

CAPÍTULO 3

ESTRUTURA DO SISTEMA JARGÃO

3.1 Introdução

O sistema Jargão gera três níveis de redes neurais evolutivas (RNE) hierarquicamente organizadas: Rede dos Conceitos Primitivos, Rede dos Conceitos Complexos e a Rede de Teorias. Estas redes são criadas contendo as informações comuns de um grupo de textos contidos na base de dados em linguagem natural (BDLN) analisada. A primeira rede consiste da Rede dos Conceitos Primitivos, que é construída por um conjunto de módulos que são associados às palavras ou associações de palavras que representam os conceitos primitivos mais significativos e freqüentes presentes nos textos. A Rede dos Conceitos Complexos é construída por um conjunto de módulos associados às frases mais significativas e freqüentes nos textos, e consiste da conexão da saída das Redes dos Conceitos Primitivos às entradas das redes de conceitos complexos. Os módulos da Rede dos Conceitos Complexos são usados como entradas da Rede de Teorias, que consistem dos textos padrões ou significativos contidos na base de dados (Figura 3.1).

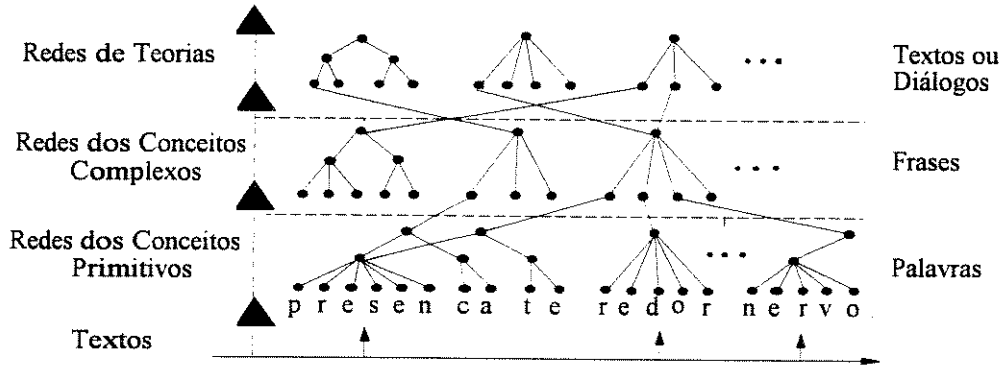

Figura 3.1 Estrutura hierárquica das RNE no sistema Jargão.

3.2 Metodologia do Sistema Jargão

A análise de uma BDLN utilizando o sistema Jargão envolve fundamentalmente os seguintes passos:

a) obtenção dos conceitos;

b) definição de uma sintaxe e sua subseqüente associação aos conceitos;

c) obtenção das combinações dos conceitos de acordo com uma sintaxe;

d) seleção e definição da semântica das combinações; e

e) recodificação dos textos da base de dados com as combinações.

Esta seqüência de passos para a utilização do sistema consiste na verdade de uma metodologia para a obtenção de informações de textos e resulta da experiência acumulada após as várias análises dos textos utilizando o sistema.

A execução desta seqüência de passos na análise de uma BDLN permite que seja alcançado o primeiro nível de extração do conhecimento, que consiste na obtenção da Rede dos Conceitos Complexos (Figura 3.1).

Para se obter um nível acima na extração do conhecimento, e uma estruturação mais complexa do conhecimento, como por exemplo, a Rede de Teorias (Figura 3.1), é necessário que seja aplicado na base de dados recodificada produzida no primeiro nível os passos descritos acima. Neste nível, é necessário que o conhecimento sintático utilizado seja, por exemplo, baseado no conhecimento da estruturação dos parágrafos e dos textos.

Portanto, generalizando, temos:

a) no nível 1 da metodologia, os passos são aplicado sobre os textos da BDLN e consistem da geração da Rede dos Conceitos Complexos;

b) no nível i + 1 da metodologia, os passos são aplicados sobre os textos recodificados no nível i e consistem da geração da Rede de Teorias.

O número de níveis alcançado na análise de alguma BDLN depende das características dos textos ou do tipo de aplicação desejada. À medida em que os níveis aumentam, o conhecimento sintático necessário para codificar as estruturas torna-se mais complexo.

3.3 Topologia das Redes no Jargão

As RNE criadas no sistema JARGÃO são compostas por módulos de três níveis: entrada, associação e saída. A conexão entre os níveis ocorre de acordo com a sintaxe especificada, podendo ser:

a) do nível de entrada para o nível de saída ou para o nível de associação;

b) do nível de associação para o mesmo nível ou para o nível de saída.

Em cada nível temos camadas de neurônios:

a) nível de entrada - formado somente por uma camada de neurônios;

b) nível intermediário ou associativo - formado por uma ou mais camadas de neurônios;

c) nível de saída - formado por somente uma camada de neurônios.

Os neurônios da camada de entrada são definidos para efetuarem sinapse com a camada associativa ou a de saída. Os neurônios das camadas associativas conectam-se com os neurônios da própria camada ou com os neurônios da camada de saída. Os neurônios das camadas de saída podem não conectar-se a outro neurônio ou conectar-se a neurônios de entrada de alguma outra rede de um nível hierárquico superior.

As estruturas da Rede dos Conceitos Complexos e Rede de Teorias são semelhantes à rede de conceitos, diferenciando-se pelo fato de poder ter mais de uma camada no nível associativo.

3.4 Obtenção da Rede dos Conceitos Primitivos (RCP)

3.4.1 Topologia da Rede dos Conceitos Primitivos

A Rede dos Conceitos Primitivos é uma RNE de três níveis com as seguintes estruturas:

a) nível de entrada - formado por somente uma camada de neurônios;

b) nível intermediário ou associativo - formado por no máximo uma camada de neurônios;

c) nível de saída: formado por somente uma camada de neurônios.

Aos neurônios da camada inferior, denominados de entradas, serão associados códigos ASCII. Os neurônios da camada de associação são de agregação, e os neurônios da camada superior são denominados de saída (Figura 3.2).

Os neurônios da camada entrada podem ser agrupados em partes:

a) a parte denominada *germe* é composta pelos neurônios associados aos caracteres iniciais, que são idênticos nas palavras incorporadas no mesmo módulo e que são agregados por um neurônio da camada intermediária. O Germe serve como índice dos conceitos;

b) a parte denominada *halo* é composta pelos neurônios associados aos caracteres que complementam o germe na formação da palavra e que são agregados por um neurônio da camada intermediária.

Um módulo ou sub-rede na Rede dos Conceitos Primitivos é criado pelo acoplamento dos neurônios da camada intermediária que agregam o germe e os halos aos neurônios da camada de saída (Figura 3.2). Portanto, num módulo pode-se ter vários neurônios de saída e a cada um deles temos associado uma palavra.

Os módulos que não possuem halo têm os neurônios do germe acoplados diretamente a um neurônio do nível de saída.

3.4.2 Configuração da Topologia da RCP

A gramática especificada para a geração da topologia da rede de conceitos simples é a seguinte:

$$V \rightarrow v_1\ v_2\ ...v_k...v_n\ C_i$$
$$C_i \rightarrow c_i\ C_{i+1}$$
$$C_i \rightarrow c_i$$

caracterizada por:

a) o símbolo inicial (germe) é uma cadeia composta de no mínimo **k** e no máximo **n** símbolos do tipo $v_1\ v_2\ ...v_k...v_n$, e que reconhece os primeiros **n** caracteres da cadeia de entrada; e

b) a presença do símbolo inicial dá início ao reconhecimento de até **m** símbolos complementares $c_1,c_2,...c_m$ (halo). Assim, o tamanho máximo **t** da cadeia reconhecida é dado por: $t= n+m$.

Portanto, os valores atribuídos para **k**, **n** e **t** definem as estruturas do germe e do halo. Em virtude deste fato, tornou-se necessário desenvolver um algoritmo que necessita desses parâmetros para a configuração da topologia da rede de conceito primitivos. Portanto, antes de efetuar a gênese da rede é necessário que sejam definidos:

a) o tamanho mínimo **k** das palavras a serem incorporadas à rede - este valor evita o reconhecimento das palavras com tamanho menor que **k**;

b) o número **n** de caracteres do germe - consiste do número mínimo de caracteres que o sistema deverá procurar similaridade entre as palavras;

c) o número máximo **t** de letras nas palavras. Através deste valor, obtém-se o número máximo de complementos **m** que compõem os halos.

Verificou-se que a correta definição desses parâmetros é importante para diminuir a complexidade do problema e aumentar a eficiência do sistema sem causar a perda das informações. Os valores sugeridos para os parâmetros acima são: tamanho mínimo das palavras igual a 3, valor do germe igual a 5 e número máximo de letras igual a 10.

Portanto, nos módulos gerados, os germes contém de 3 a 5 símbolos e os halos são compostos de até 5 símbolos. Esses valores são sugeridos porque foram utilizados com sucesso em várias análises.

As restrições as palavras pequenas (geralmente menores que 3 caracteres) devem-se ao fato de que as mesmas geralmente são preposições, conjunções e artigos, e são, em geral, pouco relevantes para os propósitos da análise. Entretanto, existem palavras pequenas (por exemplo pé, fé) que são significativas para o estudo pretendido. O sistema dispõe de mecanismos para incorporar palavras como módulos da rede de conceitos, previamente ao processo de gênese da rede. Desta forma, as palavras menores que o tamanho definido, e que são significativas para o estudo pretendido, podem ser previamente incorporadas como módulos da rede.

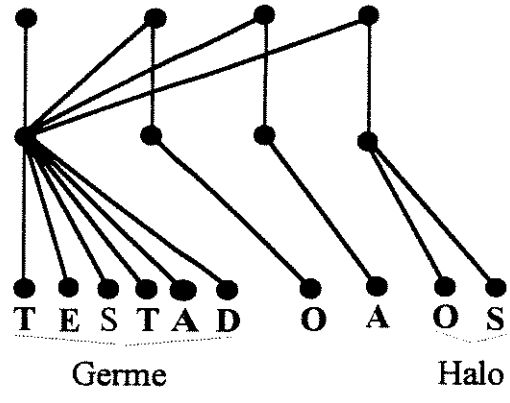

Figura 3.2: Módulo da Rede de Conceitos Primitivos

3.4.3 Gênese da Rede dos Conceitos Primitivos (RCP)

O início da geração da rede de conceito é feito através da especificação de receptores associados a caracteres do código ASCII que são atribuídos aos neurônios a serem criados no nível de entrada. Isto porque, considera-se cada palavra na BDLN como uma cadeia de transmissores, e cada caractere ASCII que compõe a palavra é um transmissor.

Assim, cada vez que uma das palavras contida nos textos é lida, percorre-se todos os módulos existentes na rede de conceitos, onde é verificada a existência de acoplamento entre os transmissores da parte da palavra denominada germe e os receptores da parte inicial da camada de entrada de algum módulo. Dependendo do resultado do acoplamento, as ações sobre os módulos da rede de conceito são feitas de acordo com as seguintes regras:

a) se não ocorrer o acoplamento com nenhum módulo existente é criado um novo módulo na rede de conceitos. No módulo, são criados tantos neurônios na

camada de entrada quanto forem as letras da palavra lida, e cada neurônio criado contém receptores para os transmissores das palavras se acoplarem. Por sua vez, os neurônios de entrada produzem transmissores para acoplarem-se com um neurônio do nível de saída. Desta forma, para toda palavra nova encontrada na base de dados um módulo novo é criado na rede de conceitos.

b) Se ocorrer o acoplamento na parte denominada germe, e ocorrer também entre a parte restante da palavra (parte denominada halo) e algum halo do módulo, então o valor das conexões sinápticas dos neurônios ativados no módulo são incrementados.

c) Se ocorrer o acoplamento na parte denominada germe, mas não ocorrer na parte restante da palavra (halo), então, no mesmo módulo da rede é incorporado um novo conjunto de neurônios com receptores para acoplarem-se com os transmissores das letras do halo da palavra. Os neurônios do halo produzem transmissores que se acoplam a um neurônio criado na camada intermediária. Os neurônios da camada intermediária que agregam o germe e o halo produzem transmissores que acoplam-se a um neurônio criado na camada de saída. Os valores das conexões sinápticas do germe são também incrementados.

3.4.4 Adaptação dos Módulos

O processo de adaptação sináptica consiste em incrementar todas as conexões sinápticas de um módulo quando acopla-se a ele a parte denominada germe e/ou também o halo de uma palavra. O valor incrementado na conexão sináptica é a_k, cujo valor é igual à ativação do neurônio pós-sináptico n_k, e é obtido pelo aumento proporcional a a_k na quantidade de transmissor no neurônio pré-sináptico n_i, e de receptor e controlador no neurônio pós-sináptico n_k. Portanto, após M acoplamentos entre n_i e n_k temos os seguintes valores para transmissor (t_i), receptor (r_j) e controlador (c_k):

$$t_i(M)=t_i(M-1)+a_k$$
$$r_j(M)=r_j(M-1)+a_k$$
$$c_k(M)=c_k(M-1)+a_k$$

Desta forma, o valor das conexões sináptica no módulo representa a freqüência de ocorrência de um conceito nos textos.

3.4.5 Sobrevivência dos Módulos

A quantidade de palavras, dependendo do tamanho e número de textos na BDLN, pode ser muito grande produzindo uma quantidade enorme de módulos. De outro lado,

temos também a restrição do tamanho da memória dinâmica do computador. Por esses fatos, podem ocorrer situações onde a quantidade de módulos gerados não é suportada pelos recursos da memória dinâmica do computador. Torna-se então necessário criar mecanismos que possibilitem a eliminação de módulos de redes sem a perda de informações importantes.

O mecanismo de sobrevivência apresentado na seção 3.4.5.1 é baseado no funcionamento do cérebro humano, onde temos uma quantidade limitada de neurônios e, evidentemente, não é possível armazenar todas as informações recebidas.

Nos três métodos implementados e mostrados a seguir, a sobrevivência dos módulos da rede de conceitos primitivos depende fundamentalmente da freqüência em que os mesmos são ativados pelas palavras contidas nos textos da BDLN.

3.4.5.1 Manipulação da Memória do Sistema

A primeira estratégia a ser apresentada refere-se à capacidade de memória dinâmica do sistema. Assim como no cérebro, são definidas três áreas de memória para a inserção dos módulos de redes, sendo as áreas denominadas de: área de curta, média e de longa duração. O tamanho de cada uma das áreas na memória dinâmica pode ser definida pelo usuário.

A manipulação das áreas do sistema é feita de forma semelhante a do cérebro. Os módulo iniciais são inseridos na área de curta duração. Quando a área de curta duração é preenchida, os módulos menos freqüentes são transferidos inicialmente para a área de média e, se necessário, para a de longa duração. Quando a área de média duração é preenchida, os módulos menos freqüentes são transferidos para a área de longa duração. Quando as três áreas de memória estão totalmente preenchidas, os módulos de baixa freqüência são eliminados das área de longa duração. Como conseqüência, os módulos mais freqüentes estão na memória de curta duração, os de freqüência intermediária na de média e os de baixa freqüência na de longa. Após a eliminação ou transferência dos módulos em uma dada área, os módulos mais freqüentes presentes na mesma área são realocados para as posições iniciais. Os módulos de redes eliminados são guardados em arquivos, permitindo ao usuário verificar se alguma das palavras eliminadas deve ser recuperada. Entretanto, após várias análises, a recuperação de algumas dessas palavras raramente ocorreu.

As vantagens desta estrutura de memória são as seguintes:

a) o sistema passou a ter as redes dispostas na seqüência temporal, evitando a eliminação de palavras em que a distribuição de ocorrência nos textos são irregulares;

b) um ganho de desempenho na busca de palavras com freqüências maiores nos módulos da rede. Isto porque os módulos mais freqüentes estão na área de curta duração e estão posicionados nas posições iniciais da memória.

3.4.5.2 Poda Automática

O segundo método de seleção é o de poda automática definida na seção 2.7.2. É aplicado após os módulos de redes terem sido construídos e tem o propósito de maximizar a capacidade de informação contida nas redes.

A entropia da rede de conceitos primitivos (RCP) criada durante o período de treinamento é dado por:

$$h(RCP) = -\sum_{i=1}^{n} Pi * \log Pi$$

onde, n é número de módulos e *Pi* é a probabilidade do módulo *i*.

A probabilidade do módulo *i* (*Pi*) é calculada pela divisão da freqüência do módulo *i* pela soma das freqüências de todos os módulos presentes na rede.

O usuário especifica o valor nebuloso β ([0,1]), que é utilizado para se obter um limiar α dado por:

$$\alpha = \beta * h(RCP)/n$$

O sistema mantém na Rede dos Conceitos Primitivos somente os módulos com valor de entropia nebulosa acima do limiar α. Os módulos dos conceitos descartados são colocados em arquivos para eventuais análises posteriores. O valor sugerido para β é 0.7, que foi obtido após várias análises de BDLN.

3.4.5.3 Poda Manual

O sistema dispõe de mecanismos que possibilitam a eliminação de módulos da rede de conceito com freqüência de ocorrência abaixo de um valor arbitrário. Este método é aplicado quando o número de módulos existentes na Rede dos Conceitos Primitivos é muito grande. Em algumas situações, a diminuição do número de módulos é necessária porque a quantia de módulos não é suportada pelo sistema, ou a maior parte dos módulos é de freqüência muito baixa, e dependendo da análise efetuada, pode ser descartada.

3.5 Geração do Dicionário

Após ter sido finalizada a geração da rede de conceito as informações nela contidas são apresentadas sob forma de uma lista (Figura 3.3). Cada elemento da lista corresponde a um módulo da RCP. A estrutura de representação de um módulo é apresentada abaixo:

GERME±PALAVRAS[21] || FREQÜÊNCIA.

A primeira parte da linha refere-se ao germe, a segunda parte, às palavras (germe concatenado com o halo) as quais o germe representa, e no final está a freqüência do módulo na rede. Os caracteres gráficos são delimitadores utilizados pelo sistema. A especificação detalhada da representação dos módulos é feita na seção 4.3.2.

Deve-se então executar a análise de cada um dos conceitos presentes na lista para:

a) verificar se existem palavras sinônimas na representação de um conceito. Se existirem, deverá ser gerada uma única representação para o conceito. Este passo consiste em juntar módulos de rede em um único, a fim de representar os conceitos que são sinônimos.

b) Gerar representações separadas de palavras que possuem mesmos germes e halos diferentes, e que possuem significados diferentes. O módulo da RCP é decomposto em tantos outros módulos quanto forem os conceitos de significados diferentes existentes. Evidentemente, em cada novo módulo o germe deve ser distinto, de tal forma que possa indexar conceitos diferentes.

c) Eliminar palavras que não tenham relação com o contexto da análise.

3.5.1 Dicionário de Conceitos Primitivos

Após manusear as redes de conceitos, obtém-se o Dicionário de Conceitos Primitivos que contém os conceitos referentes ao contexto que esta sendo enfocado. Os conceitos encontrados são inseridos em uma base de dados, onde são descritas as informações contextuais do conceito. Este dicionário pode então ser utilizado para análise de outros textos e serve também como o primeiro conjunto de informação que auxilia na estruturação e descrição do contexto trabalhado.

SAUDE±SAUDE[21] | cpt&| 793
SECRETARIA±SECRETARIA[21] | CPTloc&| 489
CENTR±CENTROS_CENTRO[21] | cpt&| 348
MUNICIP±MUNICIPIO_MUNICIPAL_MUNICIPAIS[21] | loc&| 336
DISPO±DISPOE_DISPOSITIV[21] | ATO&| 330
DOACAO±DOACAO[21] | OBJato&| 318
AUTORIZA±AUTORIZA[21] | OBJato&| 292
DENOMINACA±DENOMINACA[21] | LOCato&| 232
TRANS±TRANSFERE_TRANSFEREN[21] | OBJato&| 221
QUADRO±QUADRO[21] | obj&| 216
ESTADO±ESTADO[21] | cpt&| 209
CARGO±CARGO_CARGOS[21] | obj&| 178
IMOVEL±IMOVEL[21] | obj&| 159
PROVIDENCI±PROVIDENCI[21] | ato&| 148
PREFEITURA±PREFEITURA[21] | LOCcpt&| 147

Figura 3.3: Fragmento de um Dicionário de Conceitos Primitivos.

3.6 Geração da Rede dos Conceitos Complexos (RCC)

O primeiro aspecto a ser levado em consideração para a geração das frases é verificar a característica da base analisada, no que se refere aos tipos de expressões nela existentes. As expressões contidas nos textos podem ser:

a) descritivas - contém a descrição de símbolos; ou

b) procedural ou declarativa - descreve ações, que são geralmente representadas por verbos.

As bases de dados analisadas geralmente contêm textos descritivos ou textos procedurais. Entretanto, pode existir base de dados contendo textos onde encontram-se as duas formas de expressões. Portanto, de acordo com as características das expressões contidas nos textos analisados, deve-se inicialmente definir no dicionário quais os conceitos primitivos que são termos chaves (símbolos ou verbos). Os conceitos primitivos não classificados como termos chaves são classificados como complemento. Podem ocorrer casos de conceitos serem classificados como termo-chave (verbos ou símbolos) e complemento.

Para a obtenção das relações (conceitos complexos) existentes na base de dado em linguagem natural foram desenvolvidos duas formas:

a) classes sintáticas simples - obtenção da Rede dos Conceitos Complexos utilizando uma gramática simples (verbo ou símbolo (V)) /complemento (C);

b) classes sintáticas complexas - obtenção da Rede dos Conceitos Complexos com a utilização de uma gramática complexa.

Em ambas formas acima é necessário que sejam especificadas as classes sintáticas presentes na gramática e que determinam a topologia da RNE da Rede dos Conceitos Complexos (RCC). Em seguida, as classes sintáticas são associadas aos termos chaves e complementos. As frases contidas nos textos são usadas como conjunto de treinamento para geração, em função da gramática especificada, da topologia da Rede dos Conceitos Complexos. Portanto, a gramática deve ser especificada de forma a codificar as relações gerais (sintáticas ou estruturais) presentes na BDLN.

3.6.1 A Topologia da Rede dos Conceitos Complexos (RCC)

A Rede dos Conceitos Complexos (RCC) é construída como uma RNE de três níveis: entrada, associação, e saída. O nível de entrada é composto por uma única camada de neurônios. O nível intermediário tem uma ou mais camadas de neurônios, sendo que o número de camadas é dependente do número de encadeamentos definidos na gramática. O nível de saída tem somente uma camada.

A rede de conceitos é acoplada à Rede dos Conceitos Complexos. Os neurônio de saída da Rede dos Conceitos Primitivos serão acoplados aos neurônios da camada de entradas da Rede dos Conceitos Complexos. Este acoplamento é feito através da definição das classes da gramática correspondentes aos transmissores e receptores; e posterior atribuição dos transmissores para os neurônios de saída da Rede dos Conceitos Primitivos e dos receptores correspondentes para os neurônios da camada de entrada da RCC.

No nível intermediário, as conexões são criadas de acordo com o número de acoplamentos e da complexidade da gramática definida.

A estrutura de um módulo ou sub-rede na RCC é a seguinte :

a) ter somente um neurônio da camada de entrada, ao qual se acopla um neurônio da Rede dos Conceitos Primitivos classificado como termo-chave. Este neurônio é que caracteriza o módulo e é sempre o primeiro neurônio do módulo;

b) os demais neurônios da camada de entrada do módulo têm acoplados a eles os neurônios da Rede dos Conceitos Primitivos classificados como complemento;

c) os neurônios da camada de entrada acoplam-se aos neurônios das camadas do nível associativo. A um neurônio de uma das camadas do nível associativo pode ocorrer os seguintes acoplamentos:

- um neurônio da camada de entrada ao qual está ligado um neurônio da Rede dos Conceitos Primitivos classificado como termo-chave, e um ou mais neurônios da camada de entrada ao qual esta ligado um neurônio da Rede dos Conceitos Primitivos classificado como complemento; ou
- neurônios da camada de entrada aos quais estão ligados os neurônios da Rede dos Conceitos Primitivos classificado como complemento; ou
- neurônios de outras camadas associativas que pertençam a camadas de nível mais baixo.

Esses acoplamentos representam as diferentes associações encontradas na BDLN.

d) Os neurônios das camadas do nível associativo acoplam-se aos neurônios das outras camadas do nível associativo que pertençam às camadas de nível mais alto ou neurônios da camada de saída.

e) Cada neurônio de saída representa uma associação termo-chave e complementos encontrada na BDLN. Portanto, tem-se tantos neurônios de saída quantas forem as associações termo-chave e complementos existentes. Cada neurônio de saída corresponde a um módulo de interpretação encontrado.

3.6.2 Configuração da Topologia da Rede RCC

A codificação da sintaxe define a topologia da RCC. Na sintaxe é especificado que em um módulo existe um termo-chave e um conjunto de complementos. O número de complementos no módulo, quando não restringido pela sintaxe, tem seu valor especificado. A restrição do número de complemento é necessária, principalmente, se a forma de geração da rede da RCC adotada é a de classes sintáticas simples. Quando se adota a forma de geração da RCC utilizando-se classes sintáticas complexas, o número de complementos não é necessário ser especificado, pois é restringido pelo número máximo de encadeamentos de classes permitidas pela sintaxe definida.

Nesta fase, as frases contidas na BDLN são utilizadas como o conjunto de treinamento da RCC. O caracter padrão de segmentação de frases de treinamento é o ponto final (.). O caracter padrão de segmentação é um parâmetro que pode ser alterado para um outro caracter específico. Podem ocorrer situações na qual as frases são pobremente segmentadas. Nesta situação, a restrição do número máximo de complementos tem a função de segmentar a frase e reduzir o tamanho e a complexidade dos módulos na rede de frases.

3.6.3 Gênese da RCC com Classes Sintáticas Simples

A análise por uma gramática contendo classes simples é recomendada para aquelas situações onde não se dispõe de conhecimento profundo dos textos contidos na base ou sobre o assunto que refere-se à base. As regras sintáticas adotadas são da seguinte forma:

$$\mathbf{V} \rightarrow \mathbf{v}\,\mathbf{C}$$
$$\mathbf{C} \rightarrow \mathbf{c}\,\mathbf{C}$$
$$\mathbf{C} \rightarrow \mathbf{c}$$

Esta sintaxe é caracterizada por requerer a presença de um símbolo inicial (**V**) que dá início ao encadeamento de um número de complementos (**C**). A seguir, os conceitos do Dicionário dos Conceitos Primitivos são classificados em termo-chave e complemento. A classificação consiste em atribuir aos neurônios de saída da Rede de Conceitos Primitivos uma quantia de transmissor, por exemplo, **v** para termos chaves e **c** para os complementos; e os receptores (**V/C**) para os neurônios da camada de entrada dos módulos (Figura 3.4). Portanto, os termos chaves recebem transmissores associados aos símbolos iniciais da sintaxe e os complementos recebem transmissores associados aos símbolos complementares da sintaxe.

Após finalizar a classificação, ocorre a gênese dos módulos da RCC de acordo com o conteúdo da base de dados, e que consiste das combinações contendo um termo-chave (símbolo ou verbo) e um número de complementos que pode variar até o valor especificado.

As frases contidas na BDLN são o conjunto de treinamento da geração dos módulos das RCC. Cada frase presente na base de dados ativará conceitos da Rede dos Conceitos Primitivos, conceitos que por sua vez, liberam transmissores que conectam-se a receptores dos neurônios do nível de entrada da Rede dos Conceitos Complexos. Se o módulo ativado da rede de conceito primitivos é termo-chave e não existem neurônios da camada de entrada da RCC com receptor correspondente, é criado um novo módulo com um neurônio com o receptor correspondente. Nessas condições, o sistema gerará tantos módulos na rede RCC quanto forem os termos chaves definidos pelo usuário na Rede dos Conceitos Primitivos. Um desses módulos gerados é mostrado na figura 3.4.

No novo módulo também são criados os neurônios na camada de entrada com receptores aos quais os transmissores associados aos complementos se acoplarão. Os neurônios da camada de entrada (aos quais termo-chave e os complementos se conectam) ativados produzem transmissores que irão acoplar-se a um neurônio criado na camada associativa. Em seguida, cria-se o neurônio de saída ao qual o neurônio ativado da camada associativa se acoplará. O neurônio de saída representará a interpretação de uma frase.

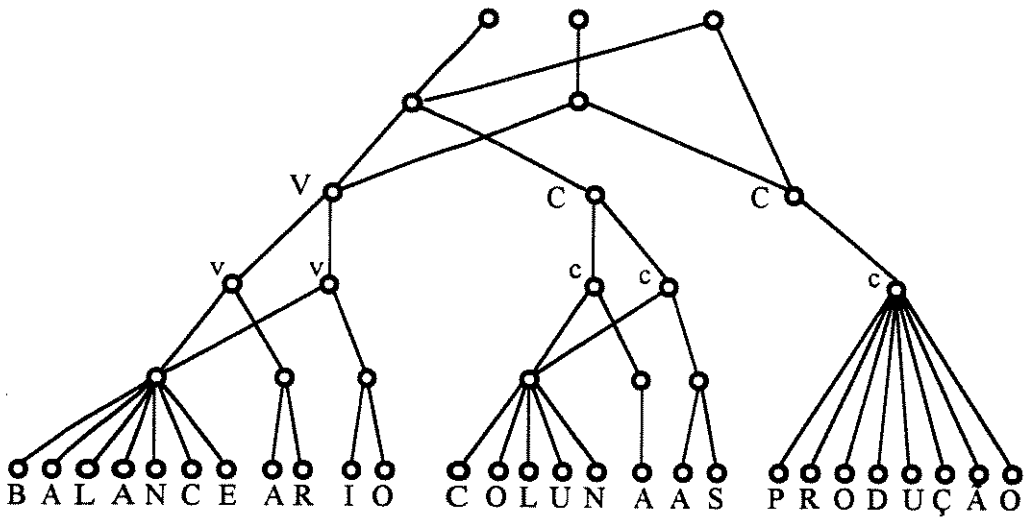

Figura 3.4: Módulo da Rede dos Conceitos Complexos com Classes Simples.

3.6.3.1 Adaptação dos Módulos

Cada frase da BDLN é exposta aos módulos existentes na rede RCC, e um deles será ativado pelo acoplamento do termo-chave do módulo com o correspondente termo-chave presente na frase. Se ocorrer o acoplamento do termo-chave e todos os complementos contidos em uma frase com algum módulo, os valores das conexões sinápticas ativadas deste módulo serão incrementados. Se o acoplamento é parcial, por não existir, por exemplo, um complemento, então cria-se um novo neurônio na camada de entrada do módulo para ser associado com o complemento, e são criadas as conexões associando o termo-chave com o conjunto de complementos para representar a frase. A parte do módulo em que ocorreu o acoplamento tem seus valores de conexões sinápticas incrementados.

O valor incrementado na conexão sináptica é a_k, cujo valor é igual a ativação do neurônio pós-sináptico n_k, e é obtido pelo aumento proporcional a a_k na quantidade de transmissor no neurônio pré-sináptico n_i, e de receptor e controlador no neurônio pós-sináptico n_k.

3.6.3.2 Sobrevivência dos Módulos

Durante a geração da rede RCC são aplicadas basicamente as mesmas técnicas de sobrevivência dos módulos que foram aplicadas à Rede dos Conceitos Primitivos. Caso os módulos presentes na RCC estejam ocupando todo o espaço de memória disponível no sistema, o algoritmo elimina os módulos menos freqüentes, ou seja, os que tenham menor

valor nas conexões sinápticas. O sistema também fornece mecanismo para a eliminação dos módulos com valor de freqüência abaixo de uma freqüência arbitrária.

Existem situações em que a informação procurada é pouco freqüente e pode ser eliminada por um dos processos acima. O sistema dispõe de um mecanismo para aumentar a importância da informação associada a um termo-chave ou complemento, conseqüentemente, o módulo gerado com essas informações terá o valor das suas conexões sinápticas (freqüência) aumentadas em proporção da sua importância. Desta forma, o módulo não correrá risco de ser eliminado nos processos de poda.

3.6.4 Gênese da Rede RCC com Classes Sintáticas Complexas

A análise lingüística nos sistemas desenvolvidos por pesquisadores da Inteligência Artificial tem sido feita utilizando analisadores sintáticos, gramática transformacional, gramática de casos, etc. O sistema Jargão tem mecanismos que permitem a utilização de conceitos lingüísticos na geração da Rede de Frases ou da Rede de Teorias. Para implementar esta abordagem são efetuados os seguintes passos:

a) definir, de acordo com o nível de conhecimento trabalhado, as classes gramaticais presentes na teoria sintática selecionada;

b) definir uma gramática (transmissores, receptores e controladores) e associar as classes presentes na teoria sintática selecionada;

c) associar aos elementos presentes no dicionário os transmissores, receptores e controladores.

Os passos acima são descritos a seguir.

3.6.4.1 Definição da Linguagem T^R>>C

Uma gramática complexa pode ser especificada e os símbolos produzidos pela gramática são associados a transmissores, receptores e controladores. A afinidade transmissor/receptor é utilizada para codificar as regras de concatenação entre as classes gramaticais em uma sintaxe. Neste contexto, deve-se definir:

a) os transmissores são representados por cadeias tais como :

- vtd, age, coa, lug, obj, ori, etc, codificando, respectivamente, verbo transitivo direto, agente, coagente, lugar, objeto, origem e destino, etc;
- suj, adj, adv, vtd, etc, codificando, respectivamente, sujeito, adjetivo, advérbio, verbo transitivo direto, etc.

b) os receptores são representados por cadeias tais como:

- VTD, VTI, AGE, COA, LUG, OBJ, ORI, etc;
- SUJ, ADJ, VTD, etc;

c) os controladores são representados por cadeias tais como:

- vtdAGE, ageCOA, vtdOBJ;
- vtdADJ, sujADJ, vtdADV;

tendo como objetivo de implementar regras sintáticas condicionais, tais como :

AGE ^ ageOBJ >> OBJ ou VTD ^ vtdAGE >> AGE; ou

SUJ ^ sujADJ >> ADJ ou VTD ^ vtdADV >> ADV.

Portanto, as cadeias da linguagem associadas aos transmissores, receptores e controladores devem representar a sintaxe que será adotada pelo JARGÃO para a análise das frases na BDLN.

Tendo sido feita a criação ou a escolha da sintaxe a ser utilizada, deve-se então:

a) atribuir diferentes classes de receptores para os termos do Dicionário de Conceitos Primitivos classificados como termo-chave, usando as cadeias correspondentes (ADJ, VTD, VTI, etc). Pode-se atribuir mais de uma classe a um termo.

b) Atribuir diferentes classes de transmissor para os termos do Dicionário de Conceitos Primitivos classificados como complementos, usando as cadeias correspondentes (suj, adj, adv, etc). Pode-se atribuir mais de uma classe a um termo.

Os termos definidos como complementos podem requerer a presença de outros complementos (SUJ, ADV, ADJ, etc). Neste caso, deve-se atribuir o controlador adequado ao complemento para implementar as regras adequadas de encadeamento. Assim, o receptor (ADJ, ADV, etc) é concatenado à cadeia de transmissores (suj, vtd, etc) contidos nos controladores (vtdADJ, sujADV, etc) a serem usados. Esse procedimento possibilita condicionar algumas das categorias de frases a serem aceitas pela Rede dos Conceitos Complexos. Por exemplo, a palavra

mao_pe_regiao/mao,pe,regiao\vtdADJ&vtiADJ

pode incorporar

anestesia_analgesico/anestesia,anestesiado,analgesico\adj&

após ter sido conectado com o verbo

provoca_causa/provocar,provoca,causa\SUJ&VTD&

que irá ser ativada pela frase

provoca anestesia nas mãos e pés.

3.6.4.2 Geração dos Módulos

As associações de símbolos da linguagem para os termos do dicionário consistem, na verdade, nas atribuições de transmissores, receptores e controladores para os módulos da Rede dos Conceitos Primitivos.

As frases contidas na BDLN são o conjunto de treinamento da geração dos módulos da RCC. Cada frase presente na base de dados ativará conceitos da rede

conceitos, que por sua vez, liberam transmissores que conectam-se aos receptores dos neurônios do nível de entrada da RCC. Se o módulo ativado da Rede de Conceitos Primitivos é termo-chave e não existem neurônios da camada de entrada com receptor correspondente, é criado um novo módulo com o receptor correspondente. Nessas condições, o sistema gerará tantos módulos na RCC quanto forem os termos chave definidos pelo usuário na Rede dos Conceitos Primitivos. Um desses módulos gerados é mostrado na figura 3.5.

No novo módulo, os neurônios na camada de entrada com receptores aos quais os transmissores associados aos complementos se acoplarão são criados somente se existir um encadeamento entre:

a) o transmissor produzido pelo termo-chave e o receptor de um complemento, ou

b) o controlador produzido por um neurônio da camada associativa, resultado de outro acoplamento, e o receptor de um complemento.

Os neurônios da camada associativa são criados para codificar os acoplamentos existentes. Portanto, ocorrendo o encadeamento, os neurônios da camada de entrada (aos quais o termo-chave e complemento acoplam-se) ativados produzem transmissores que irão acoplar-se a um neurônio criado na camada associativa. Em seguida, cria-se o neurônio de saída ao qual o neurônio ativado da camada associativa se acoplará. Deve-se salientar que os neurônios da camada associativa representam o encadeamento sintático e o neurônio de saída representará a interpretação da frase na sintaxe codificada (figura 3.6).

Portanto, o resultado consiste em:

a) criar tantos módulos na redes de conceitos complexos quanto forem o número de termos chaves definidos.

b) O termo-chave acopla-se ao neurônio mais à esquerda da camada de entrada do módulo. Esse neurônio caracteriza o módulo da Rede dos Conceitos Complexos.

c) Os outros neurônios da camada de entrada são complementos e eles produzem receptores para diferentes categorias sintáticas aceitas pelos termos chaves e seus próprios complementos. Nesse caso, são criados tantos neurônios na camada de entrada quantas forem as classes sintáticas requeridas pelos verbos e seus complementos.

d) São criados neurônios intermediários para agregarem os encadeamentos, que foram criados à partir do neurônio associado ao termo-chave com os outros neurônios da camada de entrada.

e) São criados neurônios de saída para agregarem os neurônios dos níveis e camadas inferiores, e cada neurônio de saída está associado a uma interpretação.

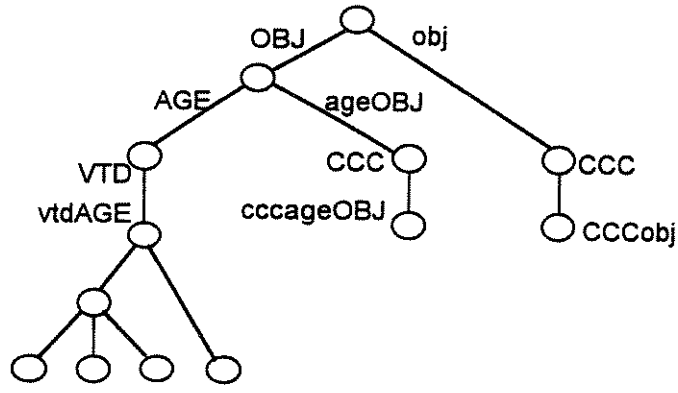

Figura 3.5: Encadeamento Sintático na Construção de um Módulo

3.6.4.3 Adaptação dos Módulos

Cada frase da BDLN é acoplada aos módulos existentes na redes de frases, tal que um deles será ativado pelo acoplamento entre o termo-chave do módulo com o correspondente termo-chave presente na frase.

Se ocorrer o acoplamento do termo-chave e de todos os complementos contidos em uma frase com algum módulo, os valores das conexões sinápticas ativadas deste módulo serão incrementados. Se o acoplamento em um módulo é parcial e existe um encadeamento (termo-chave e complementos), então cria-se:

a) os neurônios da camada de entrada do módulo para serem associados aos novos complementos existentes;

b) um neurônio de uma das camadas associativas para agregar a parte do módulo ativado (parte onde ocorreu o acoplamento) aos neurônios criados na camada de entrada. A parte do módulo em que ocorreu o acoplamento tem seus valores de conexões sinápticas incrementados.

O valor aumentado na conexão sináptica é a_k, cujo valor é igual a ativação do neurônio pós-sináptico n_k, e é obtido pelo aumento proporcional à a_k na quantidade de transmissor no neurônio pré-sináptico n_j, e de receptor e controlador no neurônio pós-sináptico n_k.

Ao acrescentarmos na geração dos módulos, por exemplo, o conhecimento sintático, provocamos a redução da explosão combinatória induzida por uma base de dados pobremente segmentada. Isto ocorre porque restringe-se a gênese das sinapses de acordo com a afinidade do transmissor/receptor. A sintaxe funciona como heurística na busca das combinações. Quanto mais restrita for a sintaxe definida, maior será seu efeito sobre a explosão combinatória.

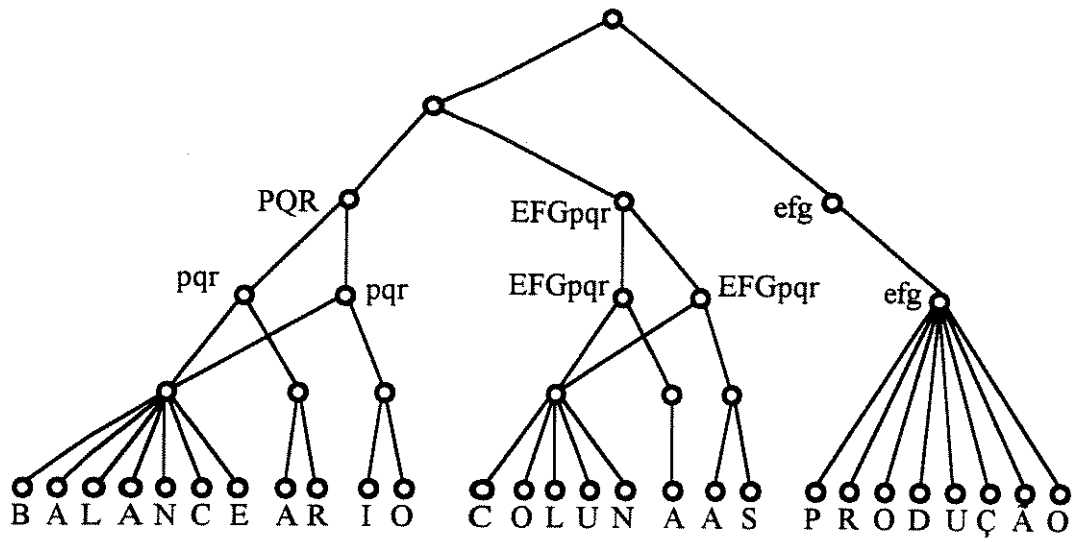

Figura 3.6: Módulo da Rede dos Conceitos Complexos com Sintaxe Complexa.

3.6.5 Avaliação das Redes Geradas

O conhecimento adquirido na RCC pode ser visualizado de duas formas diferentes: o significado específico através das frases que consistem das cadeias de palavras ou símbolos; e a descrição conceitual, através das estruturas que consistem das classes presentes nas derivações e das instâncias das classes.

As frases são construídas pelas associações termo-chave / complemento geradas de acordo com a sintaxe definida. Cada frase consiste de uma interpretação em um módulo. As interpretações existentes nos módulos da RCC são reescritas no Dicionário de Frases na forma de lista (Figura 3.7) para visualização e manipulação. Cada interpretação corresponde a uma linha da lista. A estrutura da linha é a seguinte:

TERMO CHAVE± COMPLEMENTOS±@CÓDIGO INTERNO$¨% FREQÜÊNCIA.

Onde, no campo "TERMO CHAVE" contém a parte germe da palavra especificada como termo-chave e no campo "COMPLEMENTOS" contém os complementos encontrados. No campo "CÓDIGO INTERNO" contém um símbolo associado pelo sistema à interpretação e no campo "FREQÜÊNCIA" é especificado a freqüência da interpretação. Os caracteres especiais são os separadores entre os campos.

```
CONTR_MANAGEMENT±EROSI_ERODED±@QUEST2004$¨% 106
CONTR_MANAGEMENT±EROSI_ERODED±PRICE_COSTS_COST±@QUEST2005$¨% 7
CONTR_MANAGEMENT±EROSI_ERODED±INCRE_GROWTH_GROWING±PRICE_COSTS_COST±
@QUEST2006$¨% 3
```

Figura 3.7: Leitura específica de um módulo.

Termo Chave					
CONTR_MANAGEMENT			**Sintaxe** CON		
Classe1	**Freq**				
ero	106				
Instâncias					
EROSI_ERODED					
Classe1	**Freq**	**Classe2**	**Freq**		
ero	7	cos	7		
Instâncias		**Instâncias**			
EROSI_ERODED		PRICE_COSTS_COST			
Classe1	**Freq**	**Classe2**	**Freq**	**Classe3**	**Freq**
ero	3	adj	3	cos	3
Instâncias		**Instâncias**		**Instâncias**	
EROSI_ERODED		INCRE_GROWTH_GROWING_REDUC_DECREAS		PRICE_COSTS_COST	

Figura 3.8: Leitura conceitual de um módulo.

As estruturas consistem nas seqüências de acoplamentos entre as classes sintáticas existentes em cada módulo, e para cada seqüência de acoplamentos mostra-se o termo-chave e as instâncias de cada classe sintática presente no acoplamento. As estruturas existentes nos módulos da RCC são reescritas no Dicionário de Conceitos Complexos na forma de lista, para visualização e manipulação, onde cada estrutura corresponde a um elemento da lista.

Na figura 3.8 é feita a leitura de uma RCC segundo a descrição conceitual. Mostra-se o encadeamento disparado pelo conceito chave "CONTR_MANAGEMENT", as classes presentes no encadeamento e os conceitos que são as instâncias das classe com a suas respectivas freqüências (valor da conexão sináptica).

As definições sintáticas associadas às classes permitem que sejam criadas complexas estruturas de representação do conhecimento tão poderosas quanto as tradicionais estruturas de representação do conhecimento na área da Inteligência Artificial (frames, dependência conceitual, redes semânticas).

Portanto, o resultado da geração das redes RCC deve ser analisado segundo o significado específico ou a descrição conceitual. Nesta análise deve-se verificar se os módulos de redes obtidos correspondem as informações desejadas neste nível de conhecimento. Se os módulos não correspondem, deve-se verificar de qual das seguintes fases os problemas decorrem:

a) das escolhas dos termos chaves;

b) da definição sintática; ou

c) da atribuição das classes sintáticas aos termos do dicionário.

Identificados e corrigidos os problemas, os processos de geração da RCC devem então ser refeitos.

3.7 Definições Semânticas

Na seção anterior mostrou-se que a leitura da rede RCC pode ser feita de duas formas diferentes e está relacionada a dois tipos de conhecimento: o significado específico, onde utiliza-se o conhecimento sintático e gera-se o Dicionário de Frases; e a descrição conceitual, onde utiliza-se a noção de classes e gera-se o Dicionário de Conceitos Complexos.

Assim, a forma de manipular a rede RCC depende do tipo de conhecimento desejado (específico ou conceitual) e da utilização que está sendo feita do sistema. No sistema, o procedimento para definição semântica é diferente para cada um dos tipos de conhecimento produzidos.

A fase de definição semântica corresponde à implementação do aprendizado por perguntas ou simplesmente por ouvir dizer. Deve-se salientar que grande parte do conhecimento humano é adquirido ou modificado utilizando informações que são obtidas através das perguntas ou simplesmente por ouvir dizer.

Na fase da criação do Dicionário de Conceito Simples o usuário interagiu com o sistema para refinar o dicionário. Após ter sido gerada a Rede dos Conceitos Complexos, novamente o usuário interage com o sistema, agora para definir a semântica que deverá ser associadas a um dos tipos de conhecimento (específico ou conceitual) produzido. É importante destacar que no contexto do sistema Jargão a semântica consiste de uma frase onde o usuário especifica a sua compreensão da informação analisada..

3.7.1 Definições Semânticas das Interpretações

Cada uma das cadeias de palavras ou símbolos, chamadas aqui de interpretações, aprendidas pelos módulos da RCC e que estão contidas no Dicionário de Frases, é

mostrada para que seja definida a sua semântica. Neste contexto, a semântica consiste de uma ou mais frases criadas pelo usuário que descrevem a sua compreensão da interpretação. Para auxiliar nesta definição, são mostradas também as frases presentes no texto do conjunto de treinamento que contém a interpretação analisada, e as outras interpretações presentes no mesmo módulo da RCC que possuem parte em comum com a interpretação corrente. Tendo estas informações, o usuário classifica as interpretações como:

a) interpretação muito bem formada - se os conceitos contidos na interpretação apresentada inequivocamente definem um significado específico no contexto trabalhado. O usuário insere as conjunções, preposições, artigos que eventualmente possam fazer parte das frases e que foram eliminados durante a geração da RCC, ou seleciona uma frase entre as frases de treinamento que representa a semântica da interpretação apresentada;

b) interpretação ambígua - se mais de um significado pode ser atribuído à interpretação apresentada. Neste caso, o usuário define uma frase que ele acredita ser a representação mais próxima da interpretação apresentada;

c) interpretação mal formada: é a interpretação semanticamente e sintaticamente incoerente. A interpretação apresentada deve ser eliminada.

As interpretações selecionadas em a) e b), com as respectivas definições semânticas, são reescritas no Dicionário de Frases (Figura 3.9). As interpretações mau formadas são eliminadas do Dicionário de Frases. Quando a decisão do usuário é b), são armazenadas as frases de treinamento utilizadas na decisão. Em caso de dúvidas futuras sobre o significado da expressão, pode-se mostrar o conjunto das frases de treinamento utilizadas para a definição semântica.

```
ABERT± TSR±@5CCP001$MANTER TSR 1,5 M ABERTO¨% 21
ABERT±DHSV±@5CCP002$TESTAR DHSV, ABERTURA¨% 18
APLICADO_PESO_ARRIADAS±PACKER_ PKR±@5CCP011$TESTAR PACKER COM PESO¨% 7
APLICADO_PESO_ARRIADAS± TSR±@5CCP013$LIBERAR TSR COM PESO¨% 7
APLICADO_PESO_ARRIADAS±HANGER_TUBING _TBG_ TH _ TH._ TH._T.H. ±@5CCP014
$ASSENTAR TUBING HANGER COM PESO¨% 6
 ASSENTA±HANGER_TUBING _TBG_ TH _ TH._ TH ±@5CCP021$ASSENTAR TUBING
HANGER ¨% 16
 ASSENTA±PACKER_ PKR±@5CCP022$ASSENTAR PACKER¨% 15
 ASSENTA±BUCHA±@5CCP025$ASSENTAR BUCHA¨% 4
 ASSENTA± ORIENTACAO_ORIENTADOR±BUCHA±@5CCP027$ASSENTAR BUCHA DE
ORIENTACAO¨% 3
 BALANCE±COLUNA±@5CCP028$BALANCEAR COLUNA DE PRODUCAO¨% 26
 BALANCE±DRILL_DP'S_DP'S_DP ±@5CCP029$BALANCEAR COLUNA DE PRODUCAO ¨% 10
 BALANCE±COLUNA±PRODUCAO±@5CCP030$BALANCEAR COLUNA DE PRODUCAO¨% 9
 BALANCE±TUBO_TUBING'S±@5CCP031$BALANCEAR COLUNA DE PRODUCAO¨% 8
CHECA± TSR±@5CCP034$VERIFICAR CURSO DE VEDACAO¨% 12
CHECA±CURSO_VEDACAO±@5CCP035$VERIFICAR CURSO DE VEDACAO¨% 10
CHECA±DPTT_CABO_ELETRICO±@5CCP042$VERIFICAR SINAL DPTT¨% 3
CINTA±DPTT_CABO_ELETRICO±@5CCP044$CINTAR CABO ELETRICO¨% 4
CIRCULA± TSR±@5CCP045$VERIFICAR CURSO POR CIRCULACAO¨% 10
CIRCULA±CURSO_VEDACAO±@5CCP046$VERIFICAR CURSO POR CIRCULACAO¨% 7
```

Figura 3.9: Fragmento de um Dicionário de Frase.

3.7.2 Definições Semânticas das Estruturas

Cada encadeamento entre as classes presentes nos módulos da RCC e as instâncias (termos do dicionário de conceitos) associadas a cada classe, denominada aqui de estrutura (figura 3.10), e contidas no Dicionário de Conceitos Complexos, é mostrado, devendo-se então:

a) verificar se os encadeamentos são significativos e representam conceitualmente uma informação referente ao contexto; e

b) se os termos que são instâncias das classes estão coerentes com o significado da classe.

De acordo com a análise acima, as estruturas podem ser aceitas ou recusadas. Para facilitar a análise da estrutura, mostra-se as frases dos textos nas quais a estrutura está presente e os outros encadeamentos existentes no mesmo módulo.

As estruturas aceitas são reescritas no Dicionário de Conceitos Complexos com as respectivas definições semânticas. As estruturas recusadas são excluídas do Dicionário.

Os índices (nome dos textos, índice na BDLN) dos textos onde a estrutura é encontrada são guardados juntos com as estruturas aceitas. As informações das estruturas podem ser utilizadas na construção da estrutura de casos e os índices são utilizados na indexação dos casos anteriores, podendo ser utilizadas em raciocínio baseado em casos.

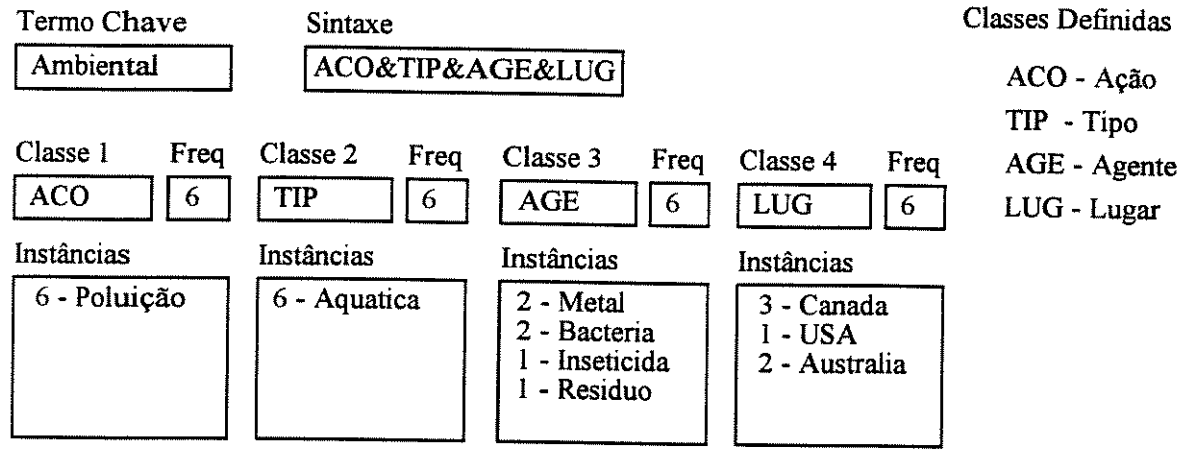

Figura 3.10: Apresentação das Instâncias de uma Estrutura.

3.8. Recodificando a Base de Dados

Após gerar a RCC e definir através da análise sintática (Dicionário de Frases) ou conceitual (Dicionário de Conceitos Complexos) a semântica restrita contida na base de dados, pode-se então reescrever a base com o novo conhecimento disponível. O sistema dispõe de mecanismo para reescrever cada um dos textos contidos na base de dados usando o conhecimento da rede RCC resultante da análise (Dicionário de Frases ou Dicionário de Conceitos Complexos). A cada uma das interpretações ou estruturas aceitas na fase de definição semântica atribui-se um código. O processo de reescrita consiste em percorrer cada um dos textos da BDLN verificando a ocorrência das interpretações ou das estruturas. Para cada texto onde verificou-se a ocorrência de interpretações, cria-se outros dois arquivos textos:

a) no primeiro escreve-se o código associado à interpretação ou à estrutura;

b) no segundo escreve-se a frase que especifica a definição semântica da interpretação ou da estrutura.

Com esta recodificação, passamos a ter uma base de dados com os textos contendo somente os códigos das interpretações ou estruturas aceitas, ou em outras palavras, contendo as informações que o usuário considera importantes no contexto analisado.

Neste momento, o usuário dispõe de um dicionário contendo os conceitos, as interpretações ou as estruturas obtidas segundo uma sintaxe e uma base de dados recodificada, que juntas proporcionam uma nova descrição da base de dados original e que podem ser utilizadas nos mais diferentes propósitos.

O sistema fornece também uma análise estatística da recodificação com as seguintes informações:

a) o histograma mostrando a distribuição dos códigos, associado a uma interpretação na base de dados;

b) a média de frases nos textos;

c) a média de frases recodificadas nos textos;

d) o número da eficácia de recuperação, que denominamos de índice de recuperação.

O histograma permite a análise quantitativa da ocorrência de conceitos complexos na BD. Os índices servem para medir a qualidade do conhecimento proporcionado pelo usuário sobre a semântica restrita.

O usuário pode, após analisar a qualidade das informações obtidas pelo sistema, decidir refazer todos ou alguns dos passos executados até esta fase. Na análise, o usuário pode descobrir falhas que tenham sido feitas na fase de definição do dicionário ou na definição semântica. Através desse processo interativo, a qualidade dos resultados tornam-se melhores, e como conseqüência, o índice de recuperação tende a aumentar. Entretanto, deve-se lembrar que o nível de recuperação depende fundamentalmente da qualidade dos textos da BDLN.

3.9 Rede de Teorias (RT)

A Rede de Teorias (figura 3.1) é uma RNE que, quando gerada, é criada para codificar o conhecimento de nível hierárquico superior ao conhecimento codificado na Rede dos Conceitos Complexos. O processo de gênese da Rede de Teorias (RT) é similar ao da gênese da RCC. O passo inicial consiste em obter um dicionário contendo os conceitos presentes na base de dados recodificada, que consiste dos símbolos associados aos neurônios de saída da Rede dos Conceitos Complexos. Em seguida, é definida uma gramática para codificar o conhecimento neste nível de RNE. O nível de complexidade da gramática a ser utilizada na gênese da RT depende fundamentalmente do nível do meta-conhecimento do conteúdo da BDLN. Neste caso, o meta-conhecimento é associado a organização dos parágrafos, textos, etc... A linguagem definida deve especificar as regras que governam as criações sinápticas na RT, ou seja, as associações entre os termos do dicionário.

Os passos executados na geração da RCC são também utilizados na geração das RT. Isto ocorre pelo fato do sistema ser construído para gerar RNE, pois o sistema foi concebido de forma que seus módulos possam ser usados na geração dos níveis de RNE.

3.9.1 Topologia da Rede de Teorias

A Rede de Teorias é construída como uma RNE de três níveis: entrada, associação, e saída. O nível intermediário tem uma ou mais camadas de neurônios, sendo que o número de camadas é dependente do número de encadeamentos definidos na sintaxe. O nível de saída tem somente uma camada.

O nível de entrada é composto por uma única camada de neurônios. A esta camada da RT é que são acoplados os neurônios de saída da Rede dos Conceitos Complexos. O acoplamento ocorre de acordo com a atribuição das classes da sintaxe correspondendo aos transmissores para os neurônios de saída da Rede dos Conceitos Complexos e receptores correspondentes para os neurônios da camada de entrada da RT.

O nível intermediário tem uma ou mais camadas de neurônios, sendo que o número de camadas e as conexões entre os neurônios são dependentes do número de encadeamentos e da complexidade da sintaxe definida. O nível de saída tem somente uma camada.

A estrutura de um módulo ou sub-rede na RT é igual a dos módulos da RCC mostrados na sessão 3.6.1.

3.9.2 Obtenção do Dicionário de Conceitos

Na criação da Rede de Teorias, inicialmente executamos os passos de obtenção de um dicionário contendo os símbolos existentes na base codificada. É importante lembrar que os símbolos presentes na BD codificada correspondem aos símbolos associados aos neurônios de saída da Rede dos Conceitos Complexos. Portanto, os termos do dicionário correspondem aos símbolos que representam os módulos da Rede dos Conceitos Complexos com a sua respectiva freqüência na BD codificada. Na geração do Dicionário de Conceitos são efetuados os mesmos passos descritos nas seções 3.4 e 3.5. Como resultado, obtêm-se os termos do dicionário ordenados pela freqüência de suas ocorrências na BD.

3.9.3 Geração da Rede de Teorias

Após a criação do Dicionário, define-se então uma gramática que governe as conexões sinápticas na RT, ou seja, que represente as relações existentes entre os termos do dicionário neste nível hierárquico de conhecimento. Na seção 3.6, mostrou-se que o uma RNE pode ser gerada de acordo com os seguintes tipos de sintaxe: simples ou

complexa, e que o tipo adotado é dependente do nível de conhecimento do contexto trabalhado. Na geração da RT, o nível da complexidade da gramática a ser utilizada depende fundamentalmente do nível do meta-conhecimento do conteúdo da BDLN que consiste da organização dos parágrafos, textos, etc... Assim, de acordo com o meta-conhecimento cria-se uma gramática simples ou complexa que especifica as relações existentes na RT.

Na geração da RT utiliza-se, por exemplo, o meta-conhecimento Tema e Rema. Cria-se uma gramática onde os símbolos do dicionário associados aos módulos da RCC mais freqüentes são escolhidos como módulo temático correspondendo ao termo-chave. Conseqüentemente, o número de módulos temáticos é que determina o número de módulos da RT. Em outras palavras, são criados tantos módulos na RT quanto forem os temas na BD. Os módulos da RCC com freqüência intermediária são escolhidos como rema. Os módulos restantes proporcionam informações complementares ao tema e rema.

Assim, de acordo com a sintaxe e a freqüência dos termos do dicionário, associam-se as classes sintáticas definidas na gramática, de forma a representar as relações que possam existir entre elas. O sistema, em função das classes definidas e das ocorrências das frases na base de dados recodificada, gera a Rede de Teorias.

3.9.4 Análise da Rede de Teorias

A análise da Rede de Teorias é feita da mesma forma que a análise da RCC mostrada na sessão 3.7, onde referencia dois tipos de conhecimento: conceitual ou específico. Assim, as interpretações (referentes ao conhecimento específico da RT) ou as estruturas (referentes ao conhecimento conceitual da RT) contidas no dicionário de Teorias são apresentadas, sendo então aceitas, se representar uma teoria coerente com o contexto analisado; caso contrário, são rejeitadas. As interpretações ou estruturas aceitas são reescritas no Dicionário de Teorias, sendo também inserida a descrição semântica correspondente.

Após a análise acima, deve-se então verificar a existência e necessidade de um outro nível de meta-conhecimento e, conseqüentemente, de uma outra RNE hierarquicamente superior. Caso isso ocorra, a base de dados, utilizada na geração da RT é recodificada, utilizando-se os termos do Dicionário de Teorias. Os passos na geração desta nova RNE são equivalentes aos descritos para a geração da RT.

Ao final da geração da RT são obtidos os textos ou diálogos mais comuns e freqüentes, segundo as sintaxes definidas durante os vários níveis e o conteúdo da BDLN.

CAPÍTULO 4

ESTRUTURA COMPUTACIONAL DO SISTEMA

4.1 Introdução

As técnicas da engenharia de software são utilizadas com sucesso no desenvolvimento de sistemas computacionais convencionais complexos. Por outro lado, para o desenvolvimento de sistemas baseados em conhecimento, as técnicas tradicionais da engenharia de software não têm produzido resultados no mesmo nível.

A análise de grande volume de textos é uma tarefa complexa, isto porque geralmente:

- cada BDLN possui uma estrutura diferente;
- cada texto possui um organização diferente; e
- cada usuário faz uma leitura, de acordo com uma semântica própria, das informações da BDLN.

O desenvolvimento de ferramentas para a análise de grande volume de textos pode ser enquadrado dentro da classe de problemas baseados em conhecimento, uma vez que a análise é feita através de algum conhecimento prévio.

Por estes motivos, não tem sido possível utilizar as técnicas tradicionais de engenharia de software para o desenvolvimento de sistema de análise de BDLN. A estratégia utilizada consiste no desenvolvimento de sistemas direcionados pela demanda ([JACO93]). Essa estratégia consiste em desenvolver inicialmente um sistema que solucione uma demanda específica. Ao surgir outra demanda, o mesmo sistema é incrementado e reavaliado, passando então a solucionar a outra demanda sem perder a capacidade de operar nas antigas.

O desenvolvimento e o aprimoramento do sistema Jargão vem sendo feito direcionado pela demanda. Os principais fatores que têm possibilitado a evolução do sistema são os seguintes:

a) a existência de uma BDLN e de profissionais que desejam obter informações nela contidas;

b) o estudo dos vários aspectos relacionados com: a organização da BDLN e dos textos; a linguagem e os formalismos de representação do conhecimento;

c) o aprimoramento dos mecanismos computacionais do sistema para suprir os requerimentos causados pelos fatores anteriores.

Um aspecto a ser salientado consiste do fato que a estrutura inicial do sistema vem sendo mantida e a sua evolução é efetuada através da incorporação dos novos procedimentos.

4.2 Estrutura do Sistema

O sistema Jargão foi desenvolvido para operar em ambiente Windows versão 3.1 ou superior. Requer equipamentos do tipo PC-386 ou superior com 4 megabits de memória. O sistema é composto de módulos e de ferramentas auxiliares. Nos módulos estão os procedimentos que compõem os estágios do sistema que são os seguintes:

Módulo Dicionário contém procedimento para a:

- geração da rede de Conceitos Primitivos;
- geração do Dicionário de Conceitos Primitivos.

Módulo Manuseio contém procedimento para a:

- atribuição da sintaxe aos elementos do Dicionário de Conceitos Primitivos;
- criação do arquivo de frases.

Módulo Combina contém procedimento para a:

- geração da Rede dos Conceitos Complexos;
- geração do Dicionário de Frases e Dicionário de Conceitos Complexos.

Módulo de Frases contém procedimento para a:

- definição da semântica das interpretações do Dicionário de Frases;
- definição da semântica das estruturas do Dicionário de Conceitos Complexos.

Módulo Recodifica contém procedimento para a:

- recodificação dos padrões (interpretações ou estruturas) nos textos;
- consolidação e geração do histograma.

Módulo Analisa contém procedimento para a:

- extração de padrões dos textos.

As ferramentas auxiliares foram criadas com o propósito de auxiliar a utilização dos módulos e são responsáveis pela:

- criação de um lote de arquivos;
- definição e seleção da sintaxe;
- ativação da ajuda;
- utilização das funções definidas no sistema Kards(Calculadora);
- ativação da base de dados acoplada ao sistema.

As interfaces do sistema foram construídas seguindo padrões do Windows. A interface de entrada é mostrada na figura 4.1, onde os módulos estão associados aos ícones da parte superior e as ferramentas auxiliares ao ícones da parte inferior. Nas seções a seguir, descreve-se cada um desses módulos e as ferramentas auxiliares.

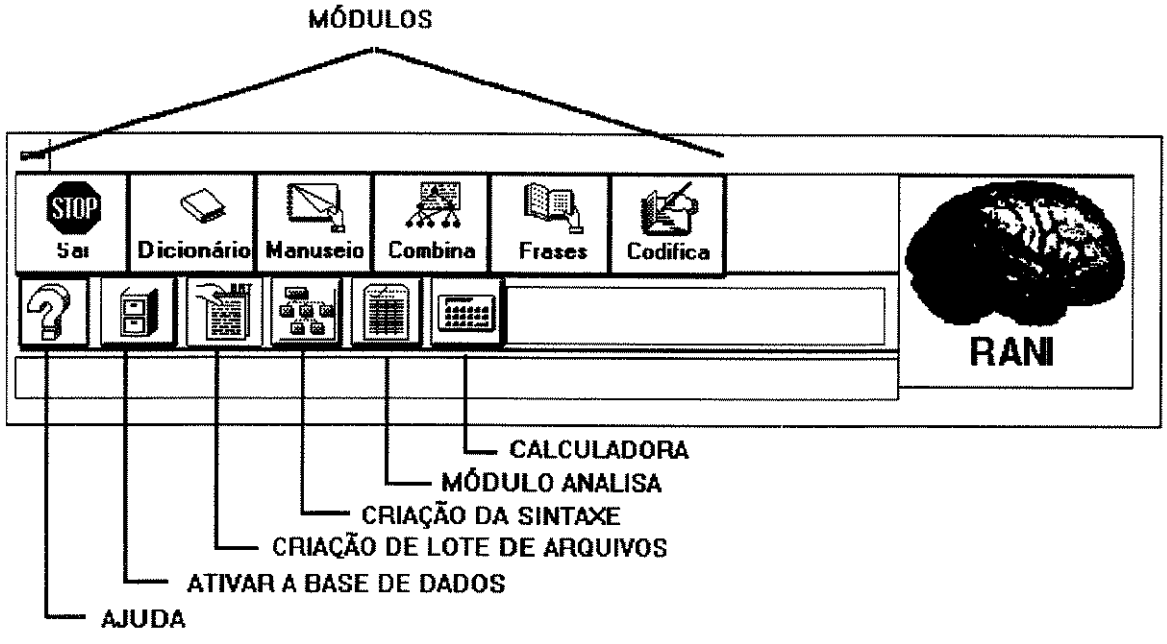

Figura 4.1: Tela inicial do Sistema Jargão.

4.3 Módulo Dicionário

Este módulo, cuja interface é mostrada abaixo, tem como função, gerar, à partir dos textos da BDLN, o Dicionário de Conceitos Primitivos (DCP). A geração é feita em duas fases: Textos e Lista. A fase Textos tem como função a geração da rede de conceitos primitivos. A fase Lista tem como função gerar o dicionário. A descrição dos procedimentos implementados nessas fases estão descritos na seção 3.4 (Obtenção das

Redes de Conceitos Primitivos) do capítulo anterior. Antes de iniciar a execução dessas fases é definido no arquivo de configuração do sistema os seguintes parâmetros:

- caractere separador de frases;
- limiar nebuloso.

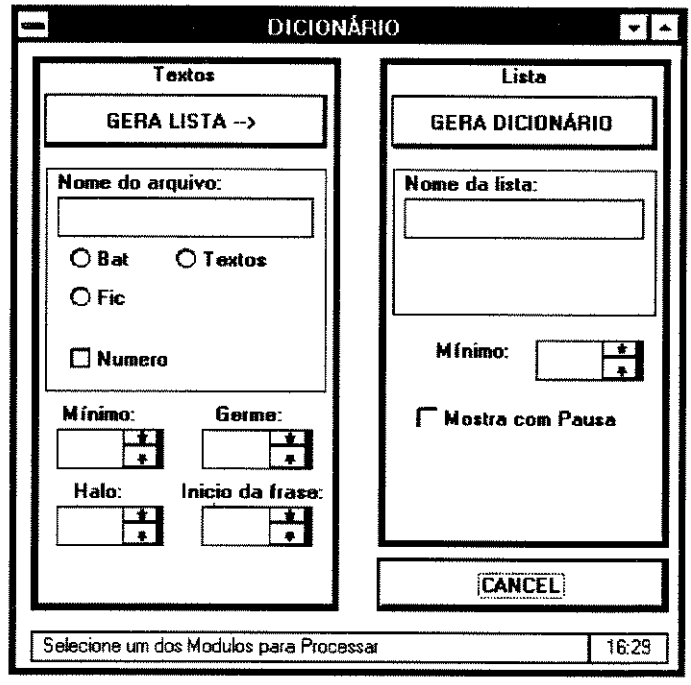

Figura 4.2: Interface do Módulo Dicionário.

4.3.1 Fase Textos

Nesta fase, à partir da BDLN, são criadas as redes de conceitos primitivos, como mostrado na figura 4.3. Esta fase é ativada ao ser selecionado o botão "GERA LISTA" na interface mostrada na figura 4.2. Em seguida, nesta interface os seguintes parâmetros são fornecidos:

- a base de dados ou os textos a serem trabalhados (Nome do Arquivo);
- as informações referentes à topologia da rede que consistem: do tamanho máximo das palavras, do germe, do halo e do início da frase;
- o nome a ser dado ao arquivo onde deve ser inserido às redes.

Os módulos criados são inseridos em uma lista e em seguida são inseridos em três arquivos com o mesmo nome porém com extensão diferente. Os arquivos criados são os seguintes:

- arquivo contendo os módulos das redes aceitas por terem valor de entropia nebulosa acima do limiar nebuloso;

- arquivo contendo os módulos das redes recusadas por terem o valor de entropia nebulosa abaixo do limiar nebuloso;

- arquivo contendo os módulos das redes eliminadas por ter esgotado a capacidade de memória dinâmica do sistema.

A representação de cada módulo da rede corresponde a uma linha do arquivo, e é construída de acordo com a seguinte sintaxe:

<módulo> → <germe>, '±.....:', <freqüência>, ' ', <tamanho do germe>, '°', <halos>,
'Entropia: ', <entropia>
<germe> → <letras>
<freqüência> → <inteiros>
<tamanho do Germe> → <inteiros>
<halos> → '[' , <início>, '/', <halo>, '](', <freqüência>, ')', <halos>
<halos> → '[', <início>, '/', <halo>, '](', <freqüência>, ')'
<halo> → <letras>
<início> → <inteiros>
<letras> → <letra> | <letra>, <letras>
<inteiros> → <inteiro> | <inteiro>, <inteiros>
<entropia> → <real>
<letra> → A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|X|Y|W|Z
<inteiro> → 1|2|3|4|5|6|7|8|9|0

Assim, o módulo da rede representada na linha
INCRE±.....:295 5° [6/ASING](38) [6/ASED](163) [6/ASES](28) [6/ASE](62) [6/MENTS](3)[6/ASED;](1) Entropia: .22
codifica as seguintes informações:

GERME: INCRE Freqüência : 295 Tamanho do Germe: 5
Halo: ASING Freqüência : 38
Halo: ASED Freqüência : 163
Halo: ASES Freqüência : 28
Halo: ASE Freqüência : 62
Halo: MENTS Freqüência : 3
Halo: ASED; Freqüência : 1
Entropia: 0.22

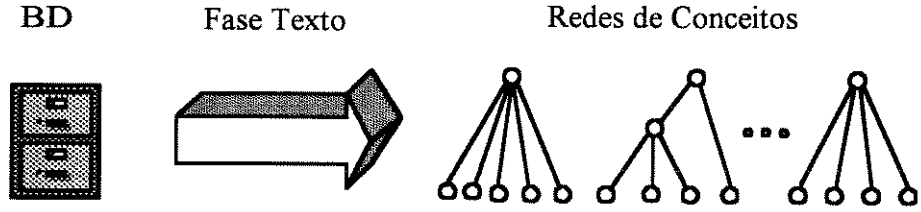

Figura 4.3: Esquema da Fase Texto

4.3.2 Fase Lista

Esta fase é ativada ao selecionar o botão "GERA DICIONÁRIO" na interface mostrada na figura 4.2. Nesta fase, pode-se eliminar os módulos da rede de conceitos primitivos que estejam abaixo de um certo valor de freqüência (especificado no campo Mínimo da interface). Os módulos da rede de conceitos primitivos remanescentes são escritos no DCP na forma de uma lista, como mostrado na figura 4.4. Cada elemento da lista corresponde a um módulo da rede que é especificado como uma cadeia de caractere de acordo com a seguinte sintaxe:

<rede> → <germe>, '±', <palavras>, '²¹ ||', <freqüência>
<germe> → <letras>
<palavras> → <palavra> | <palavra>, '_', <palavras>
<palavra> → <letras>
<letras> → <letra> | <letra>, <letras>
<freqüência> → <inteiros>
<inteiros> → <inteiro> | <inteiro>, <inteiros>
<letra> → A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|X|Y|W|Z
<inteiro> → 1|2|3|4|5|6|7|8|9|0

Na linha abaixo é apresentado um elemento do dicionário

DESCID±DESCIDO_DESCIDA_DESCIDAS²¹ || 82

contendo as seguintes informações:

GERME: DESCID
PALAVRAS: DESCIDO_DESCIDA_DESCIDAS
FREQÜÊNCIA: 82

Os caracteres especiais são os separadores entre os campos.

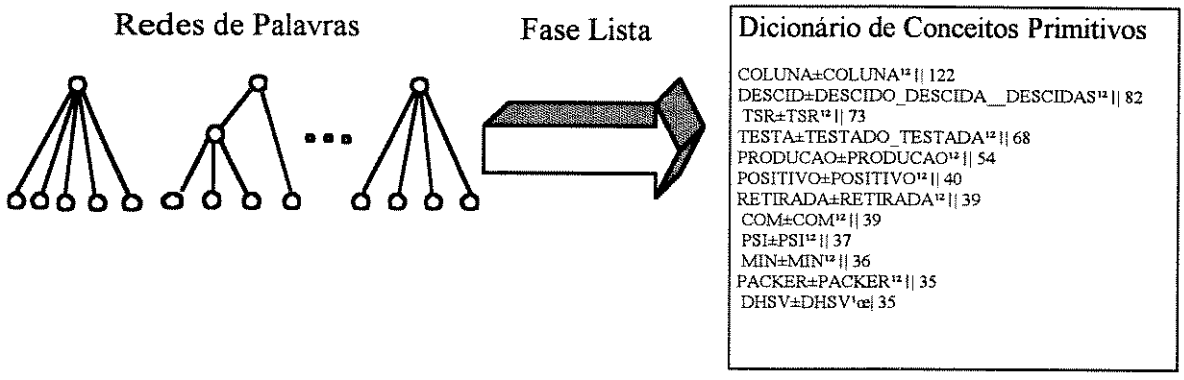

Figura 4.4: Esquema da Fase Lista.

4.4 Módulo Manuseio

Este módulo, cuja interface é mostrada na figura abaixo, é composto por duas fases. A fase Manuseio é utilizada para a manipulação e atribuição dos símbolos da sintaxe aos termos do dicionário de conceitos primitivos. A fase Separa Frase é executada após ter sido criado o dicionário de conceitos primitivos; e para cada termo do dicionário definido como termo chave, cria-se um arquivo contendo todas as frases onde ele esteja presente.

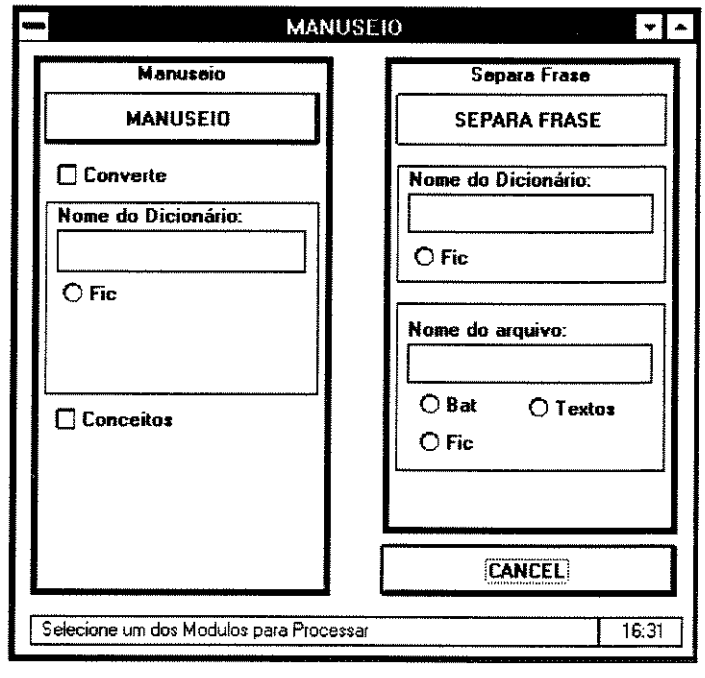

Figura 4.5: Interface do Módulo Manuseio.

A utilização deste módulo é feita após ser criada a sintaxe que será atribuída aos termos do DCP. A interface para inserir a sintaxe no sistema é mostrada na seção 4.9.1. A seguir, descreve-se detalhadamente cada uma das fases.

4.4.1 Fase Manuseio

Esta fase é ativada ao selecionar o botão "MANUSEIO" na interface mostrada na figura 4.5. Ela foi implementada para o manuseio da lista que contém os módulos da rede. O intuito desta fase é por exemplo, criar o dicionário de conceitos primitivos, como descrito na seção 3.5. A utilização desta fase pode ser feita de duas formas distintas: através do significado específico, que consiste em operar o sistema utilizando o conhecimento específico do usuário; ou pela descrição conceitual que consiste em operar o sistema utilizando a noção de classe. A utilização padrão é feita pelo conhecimento específico. A utilização pela descrição conceitual é feita quando é selecionado o item "Conceitos" na interface mostrada na figura 4.5.

Para cada uma das formas acima implementou-se uma interface. Na interface para operar com o significado específico (figura 4.6) é fornecida a lista dos módulos da rede. Nesta interface o usuário pode executar uma das seguintes operações sobre a lista: buscar uma dada cadeia de caractere na lista (ativado pelo botão BUSCA); incrementar a freqüência de um módulo (FREQÜÊNCIA); agrupar os módulos que referem a termos sinônimos (SINÔNIMO); eliminar os módulos da rede; e a atribuição dos símbolos da sintaxe definida aos módulos da rede (SINTAXE).

Na interface para operar com a descrição conceitual (figura 4.7) mostra-se também a lista dos módulos e a sintaxe. Entretanto seu modo de operação é diferente da interface anterior. Isto porque, busca-se inicialmente encontrar na lista o termo chave que é inserida no campo "VERBO". As classes sintáticas associadas ao termo chave são especificadas (campos CLASSE1, CLASSE2, ...), e em seguida, são selecionados na lista as possíveis instâncias dessas classes.

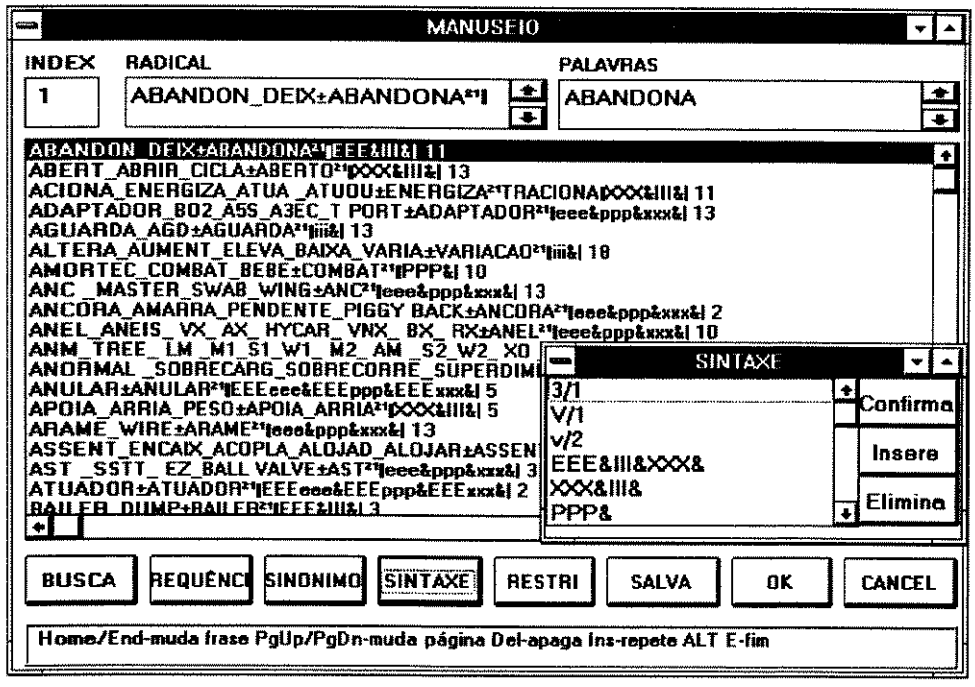

Figura 4.6: Interface para Atribuição Sintática.

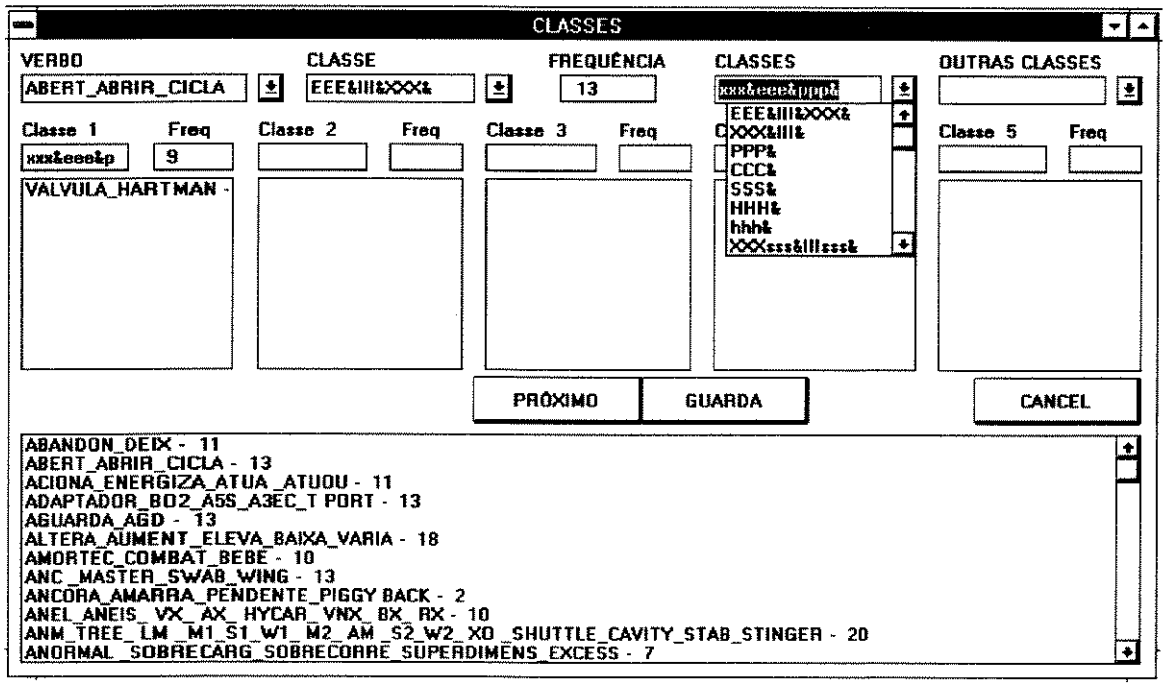

Figura 4.7: Interface para Definição das Classes.

4.4.2 Fase Separa Frase

Esta fase é ativada ao selecionar o botão "SEPARA FRASE" na interface mostrada na figura 4.5. Nesta fase, é criado um conjunto de arquivos contendo frases. Os arquivos são criados para todos aqueles conceitos do dicionário definidos como termo chave. O nome do arquivo corresponde ao termo chave, e nele são inseridas todas as frases da BDLN onde o termo chave ocorre.

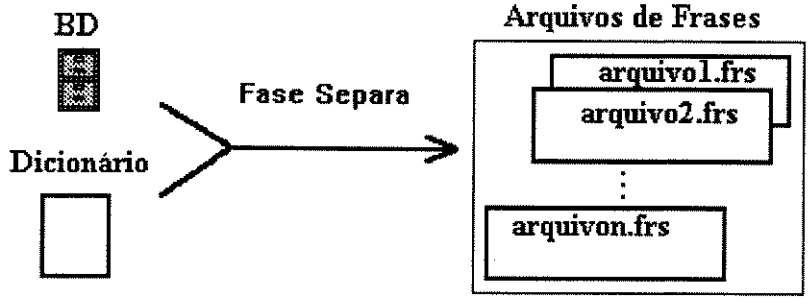

Figura 4.8: Fase Separa Frase.

4.5 Módulo Combina

Neste módulo, cuja interface é mostrada na figura 4.9, são criados os módulos de uma RNE complexa de acordo com: a sintaxe definida e atribuída aos termos do dicionário de conceitos; e com o conteúdo da BD. Juntamente com a RNE complexa são geradas as instâncias das classes sintáticas na RNE. A criação das redes é feita em duas fases seqüênciais: Fase de Cluster e Fase de Frases. A seguir mostra-se a descrição de cada uma das fases deste módulo.

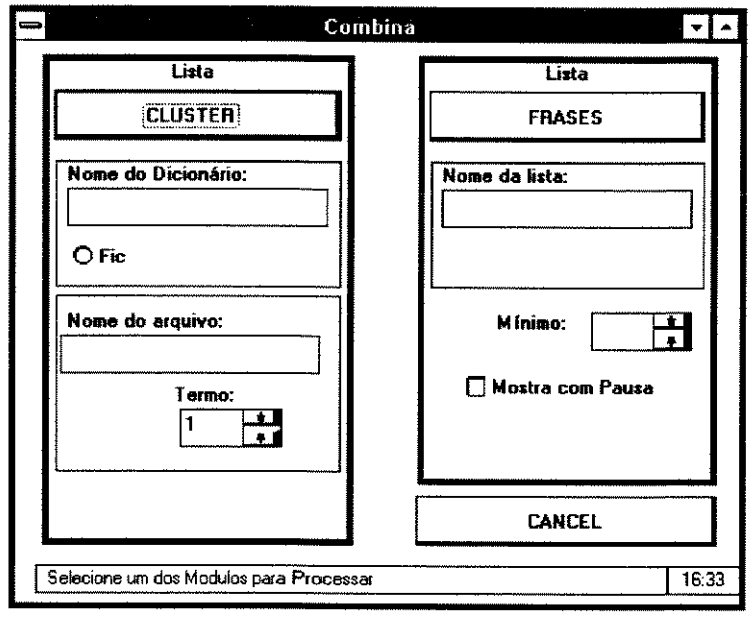

Figura 4.9: Interface do Módulo Combina.

4.5.1 Fase Cluster

Esta fase é ativada ao selecionar o botão "CLUSTER" na interface mostrada na figura 4.9. Nela, é gerada a RNE complexa e as instâncias das classes, de acordo com: a gramática criada pelo usuário; a atribuição dos símbolos da sintaxe aos termos do Dicionário de Conceitos Primitivos; e com o conteúdo da BD (Figura 4.10).

O algoritmo implementado nesta fase pode gerar as redes de acordo com duas formas de codifição sintática:

- sintaxe com uma gramática simples;
- sintaxe com uma gramática complexa.

Os módulos da rede gerada e as instâncias das classes nos módulos da rede são inseridas em arquivos distintos, cujos nomes são definidos pelo usuário. Cada módulo da rede é armazenado em uma linha de um arquivo, conforme a seguinte estrutura:

<módulo> → <germe>, '±', <palavras>, '[21] |', <sintaxe >, '|.....',
<lista dos agrupamentos>
<germe> → <letras>
<palavras> → <palavra> | <palavra>, '_', <palavras>
<palavra> → <letras>
<sintaxe > → <letras>, '&' | <letras>, '&', <sintaxe>
<lista dos agrupamentos> → '{', <termo chave>, '}{', <lista complemento>, '}'
<termo chave> → '[/', <índice termo chave>, '] (', <freqüência>, ') '

<lista complemento> → <complementos> | <complementos>, <lista complemento>
<complementos> → '[/', <índice complementos>, '](', <freqüência>, ')'
<índice complementos> → <índice> | <índice>, '/', <índice complementos>
<índice> → <inteiros>
<letras> → <letra> | <letra>, <letras>
<freqüência> → <inteiros>
<inteiros> → <inteiro> | <inteiro>, <inteiros>
<letra> → A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|X|Y|W|Z
<inteiro> → 1|2|3|4|5|6|7|8|9|0

As classes e as instâncias existentes em cada módulo da rede são armazenadas em uma linha de um arquivo, de acordo com a seguinte estrutura:

<linha> → <germe>, '±', <palavras>, '21 |',
<sintaxe >, '|....:', <freqüência>, <lista dos agrupamentos>
<germe> → <palavras>
<sintaxe > → <letras>, '&', <sintaxe>
<sintaxe > → <letras>, '&'
<palavras> → <palavra> | <palavra>, '_', <palavras>
<palavra> → <letras>
<lista dos agrupamentos> → <lista das classes>, <lista dos agrupamentos>
<lista dos agrupamentos> → <lista das classes>
<lista das classes> → <classes>, '{', <lista dos termos>, '}(', <freqüência>, ') '
<classes> → '[/', <classe>, ']'
<classe> → <letras> | <letras>, '/', <classe>
<lista dos termos> → '/', <índice do termo>, '\', <freqüência>, <lista dos termo>
<lista dos termos> → '/', <índice do termo>, '\', <freqüência>
<índice termo> → <inteiros>
<letras> → <letra> | <letra>, <letras>
<freqüência> → <inteiros>
<inteiros> → <inteiro> | <inteiro>, <inteiros>
<letra> → A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|X|Y|W|Z
<inteiro> → 1|2|3|4|5|6|7|8|9|0

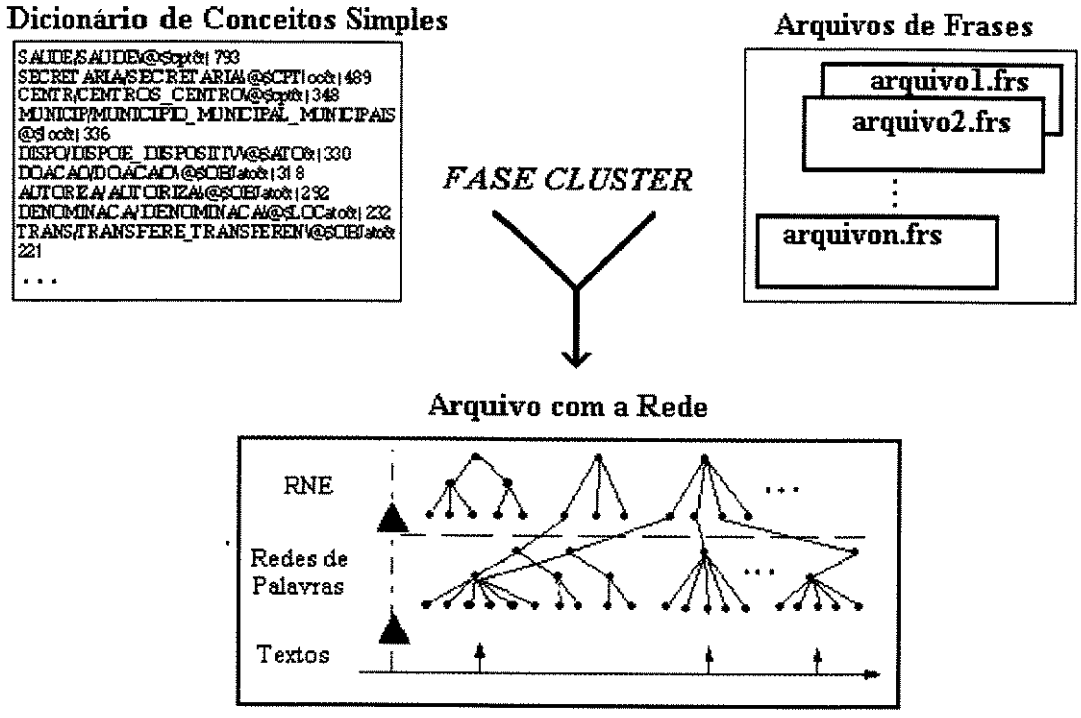

Figura 4.10: Esquema da Fase Cluster.

4.5.2 Fase Frase

Esta fase é ativada ao selecionar o botão "FRASE" na interface mostrada na figura 4.9. Nela, pode-se eliminar as interpretações contidas na rede que têm freqüência abaixo de um certo valor. As interpretações remanescentes são reescritas em um arquivo, na forma de lista, gerando o Dicionário de Frases. Cada elemento da lista corresponde a uma interpretação que é construída com a seguinte estrutura:

<frase> → <termo chave>, '±', <complementos>, '@', <código>, '$¨%', <freqüência>
<termo chave> → <palavras>
<palavras> → <palavra> | <palavra>, '_', <palavras>
<palavra> → <letras>
<complementos> → <palavras>, '±' | <palavras>, '±', <complementos>
<código> → <letras>, <inteiros>
<freqüência> → <inteiros>
<letras> → <letra> | <letra>, <letras>
<inteiros> → <inteiro> | <inteiro>, <inteiros>
<letra> → A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|X|Y|W|Z
<inteiro> → 1|2|3|4|5|6|7|8|9|0

Assim, a linha

ABERT± TSR±@5CCP001$¨% 21

tem as seguintes informações:

Termo Chave: ABERT

Complemento: TSR

Código da interpretação: 5CCP001

Freqüência: 21.

Os caracteres especiais consistem dos separadores.

As associações entre as classes sintáticas e as instâncias de cada classe presentes nas associações encontradas nos módulos da RNE são também reescritas em arquivo gerando o Dicionário de Conceitos Complexos. Abaixo mostra-se um fragmento do arquivo de classes gerado. No módulo ABERT a classe sintática associada ao verbo é a MNO&. Os complementos presentes no módulo associado a classe mno& com as respectivas freqüências são apresentadas.

Verbo : ABERT Sintaxe: MNO& Freqüência: 9

CLASSE: mno

ELEMENTOS DA CLASSE mno&

TSR Freqüência: 6

CURSO_VEDACAO Freqüência: 2

SHEAR_OUT_SEDE Freqüência: 1

CAMISA Freqüência: 1

HANGER_TUBING _TBG_ TH _ TH,_ TH._T.H. Freqüência: 2

DHSV Freqüência: 2

LC_L.C. Freqüência: 2

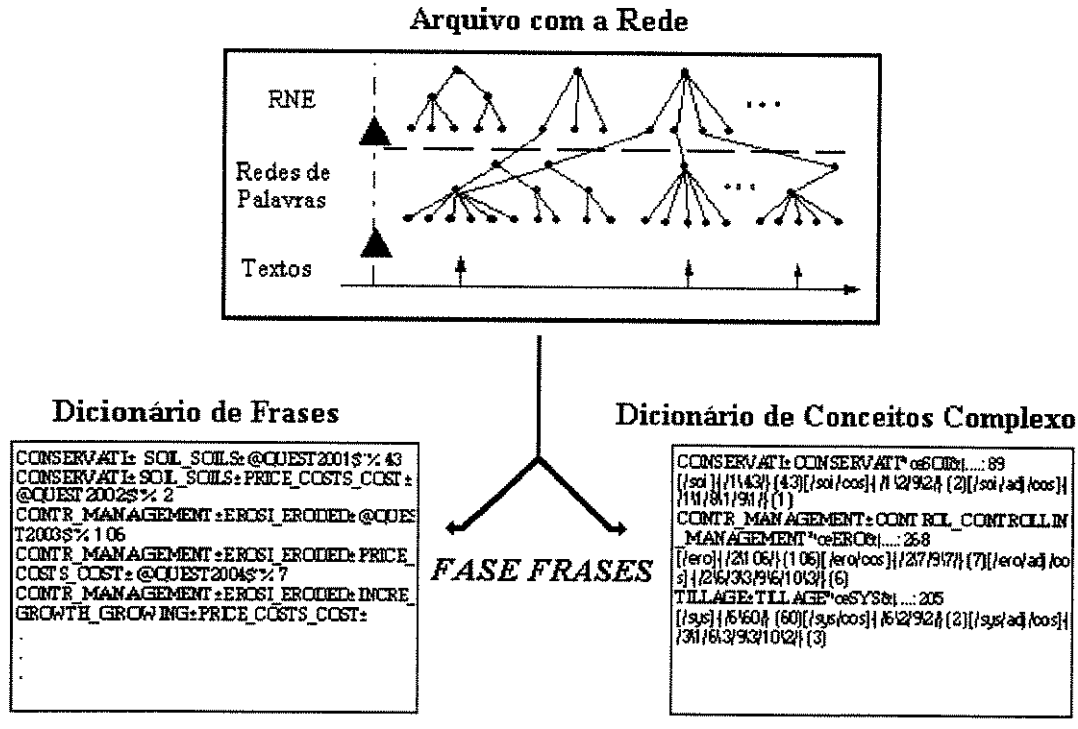

Figura 4.11: Esquema da Fase Frases.

4.6 Módulo Frases

Este módulo é usado para a análise e a definição semântica das interpretações contidas no Dicionário de Frases ou das estruturas contidas no Dicionário de Conceitos Complexos.

A fase Definição de Frases é usada para a análise e definição semântica das interpretações contidas no Dicionário de Frases. A utilização desta fase ocorre quando a análise é feita adotando-se o conhecimento lingüístico.

A fase Analisa Classes é usada para a análise das estruturas contidas no Dicionário de Conceitos Complexos. A sua utilização ocorre quando a análise é feita segundo uma abordagem conceitual.

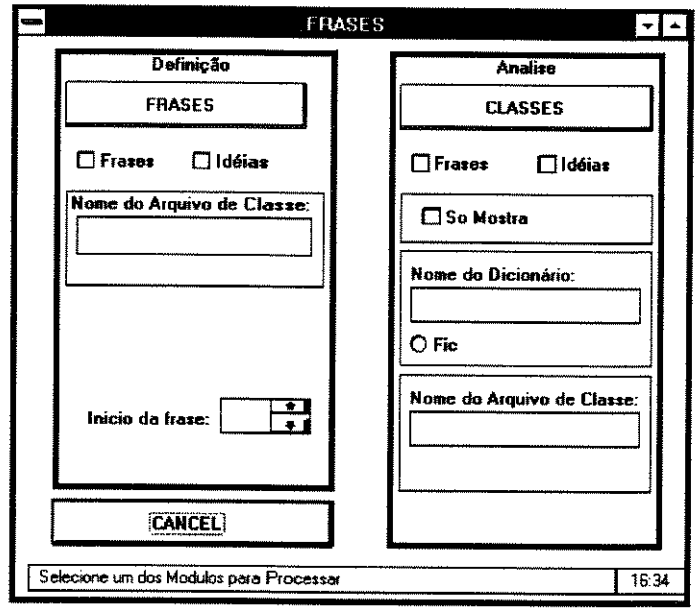

Figura 4.12: Módulo Frase.

4.6.1 Fase de Definição de Frases

Esta fase é ativada ao selecionar o botão "FRASES" na interface mostrada na figura 4.12. Ela é ativada para a análise das interpretações presentes na RNE, e que estão no Dicionário de Frases. As interpretações que são consistentes com o contexto trabalhado são aceitas e recebem uma definição semântica correspondente. A análise é feita através de uma interface (Figura 4.13), onde mostra-se: a interpretação; as frases contendo a interpretação e as outras interpretações do módulo que são similares. Assim, após o término da análise das interpretações, o Dicionário de Frases é refeito, mantendo-se somente as interpretações aceitas com as correspondentes definições semânticas. A seguir, mostra-se como é estruturada uma interpretação no Dicionário de Frases:

<frase> → <termo chave>, '±', <complementos>, '@', <código>, '$', <semântica>
'%', <freqüência>

<termo chave> → <palavras>

<palavras> → <palavra> | <palavra>, '_', <palavras>

<palavra> → <letras>

<complementos> → <palavras>, '±' | <palavras>, '±', <complementos>

<código> → <letras>, <inteiros>

<semântica> → <palavra> | <palavra>, ' ', <semântica>

<freqüência> → <inteiros>

<letras> → <letra> | <letra>, <letras>

<inteiros> → <inteiro> | <inteiro>, <inteiros>

<letra> → A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|X|Y|W|Z
<inteiro> → 1|2|3|4|5|6|7|8|9|0

Assim, na interpretação

ASSENTA±PACKER_ PKR±@5CCP022$ASSENTAR PACKER¨% 15

temos:

Termo chave: ASSENTA
Complemento: PACKER_ PKR
Código da Interno: 5CCP022
Semântica Definida: ASSENTAR PACKER
Frequência: 15

As RNE podem ser acopladas em níveis hierárquicos (seção 2.6). Isto ocorre quando a RNE_i acopla-se com a RNE_{i+1}. Esta fase é utilizada para a análise dos vários níveis de RNE geradas. Assim, quando pretende-se analisar as RNE do nível de conceitos complexos, ou seja, RNE do primeiro nível, é selecionado na interface acima o campo Frases. Para análise da RNE dos níveis seguintes deve ser selecionado o campo Idéias. A diferenciação é necessária em virtude dos dicionários a serem utilizados, uma vez que:

- na análise da RNE de frases (Frases) é fornecido apenas o Dicionário de Frases (D_1);

- na análise da RNE do nível i+1 (Idéias) são necessários os dicionários D_i e D_{i+1}.

As informações contidas no Dicionário de Frases podem ser inseridas em uma base de dados.

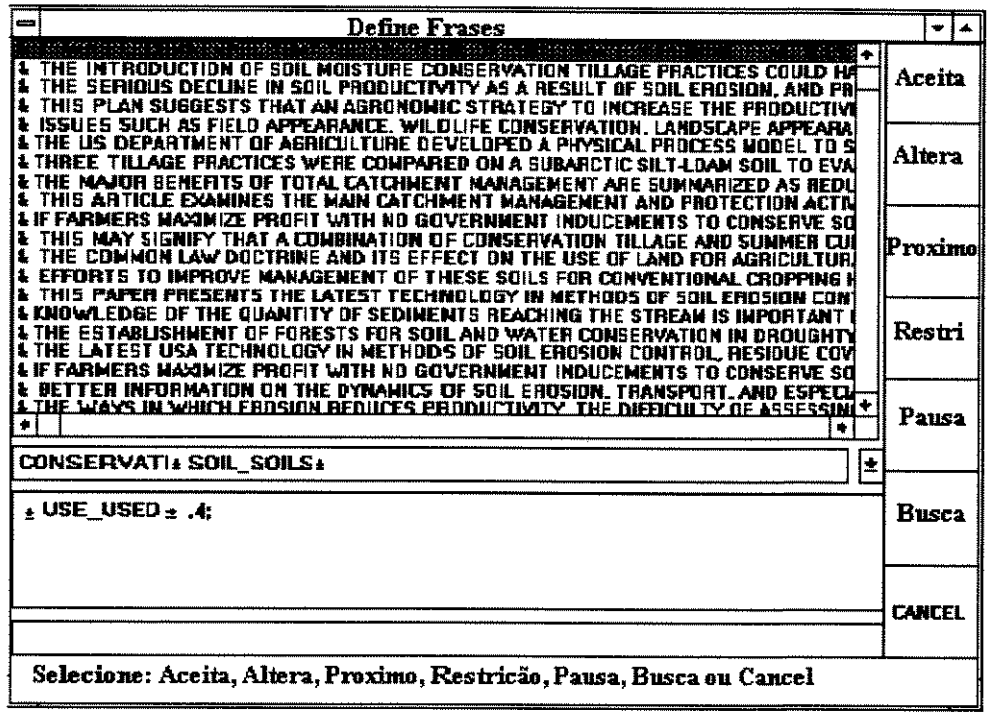

Figura 4.13: Interface para a definição semântica das interpretações do Dicionário de Frases.

4.6.2 Fase de Análise de Classes

Esta fase é ativada ao selecionar o botão "CLASSES" na interface mostrada na figura 4.12. Nela, é executada a análise das estruturas contidas no Dicionário de Conceitos Complexos que consistem das associações entre as classes presentes nos módulos da RNE e os termos que são as instâncias das classes. Para efetuar a análise da estrutura, criou-se uma interface (Figura 4.14), onde mostra-se: o termo chave (campo VERBO); as classes sintáticas encadeadas (Classe1, Classe2, ...) e os termos que são as instâncias das classes; e as frases contendo os elementos das instâncias. As estruturas consistentes com o contexto trabalhado recebem uma definição semântica correspondente. Assim, após o término da análise das associações, o Dicionário de Conceitos Complexos é refeito, mantendo-se somente as classes e as instâncias aceitas com as correspondentes definições semânticas. Abaixo temos uma instância de uma classe aceita:

USE_ USED_USING_APPLI±FERTIL±INCRE@ USE&use&inc$ USE&use&inc"%4; 4;4

onde representa:

	Classe	Frequência
Termo Chave: USE_ USED_USING_APPLI	USE&	4
Instância 1: FERTIL	use&	4
Instância 2: INCRE	inc&	4
Código Semântico: USE&use&inc		

Esta fase é utilizada para a análise das classes das RNE de diferentes níveis. Quando pretende-se analisar as classes de uma RNE do nível de Conceitos Complexos, ou seja, RNE do primeiro nível, deve ser selecionado na interface acima o campo Frases. Para análise das classes de RNE dos níveis seguintes, como Rede de Teorias, deve ser selecionado o campo Idéias.

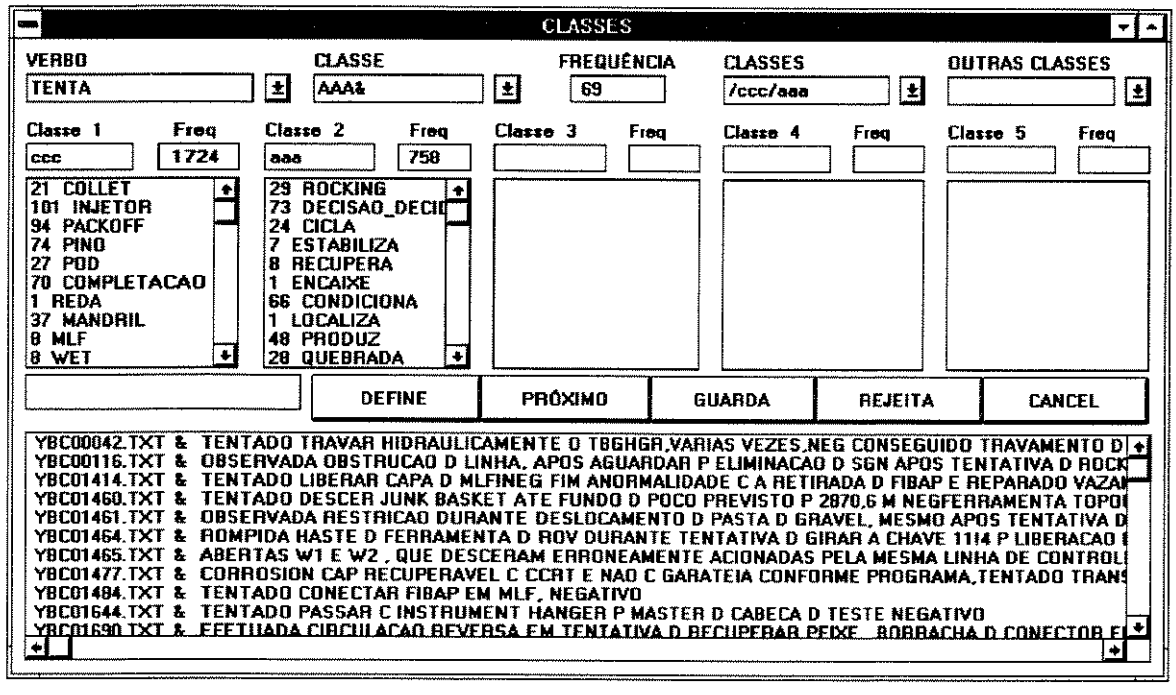

Figura 4.14: Interface para a definição semântica das classes do Dicionário de Conceitos Complexos.

Nesta fase, os índices da BD onde estão as frases que fornecem suporte para as estruturas escolhidas são armazenados em arquivo (arquivo de índice). Assim, tem-se associado ao conhecimento conceitual contido nas estruturas, os índices da BD, e essas

informações são conjuntamente utilizadas na implementação do raciocínio baseado em casos.

As informações contidas no Dicionário de Conceitos Complexos e no arquivo de índices podem ser inseridas em uma base de dados.

4.7 Módulo Codifica

A base de dados analisada é recodificada com o conhecimento contido na RNE (Dicionário de Conceitos Complexos ou Dicionário de Frases), resultando na criação de uma nova base de dados. Como conseqüência, a nova base de dados conterá somente o conhecimento consistente com o contexto em que se está trabalhando. Na figura 4.15 é apresentada a interface deste módulo. As fases que compõem este módulo são Recodifica e Consolida, devendo ser executadas na seqüência em que são apresentadas.

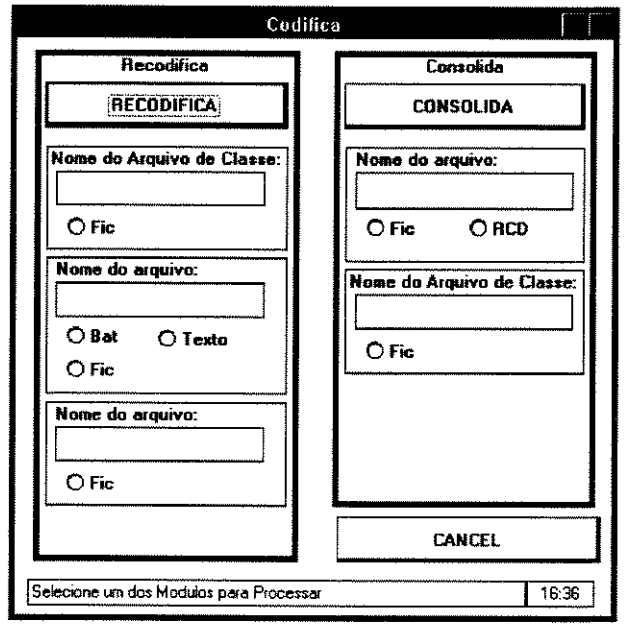

Figura 4.15: Módulo Codifica.

4.7.1 Fase Recodifica

Esta fase é ativada ao selecionar o botão "CODIFICA" na interface mostrada na figura 4.15. Nesta fase os arquivos da base de dados utilizada são recodificados utilizando o conhecimento contido na RNE gerada neste nível de conhecimento

processado. Como conseqüência é gerado uma nova base de dados. Portanto, a geração da nova base de dados codificada é feita utilizando:

- o arquivo contendo as interpretações ou as estruturas relativas a RNE produzida na fase anterior (Dicionário de Conceitos Complexos ou Dicionário de Frases);

- a base de dados que está sendo analisada.

Em cada um dos arquivos da BD analisada, é verificada a ocorrência do conhecimento da RNE (interpretações ou as classes com as instâncias). Quando a presença é verificada, os códigos correspondentes são escritos na nova BD. Conseqüentemente, a nova base de dados tem a mesma estrutura da base que está sendo analisada.

4.7.2 Fase Consolida

Esta fase é ativada ao selecionar o botão "CONSOLIDA" na interface mostrada na figura 4.15. Nesta fase é feita para a eliminação das informações redundantes contidas na nova BD. Ao mesmo tempo, é gerado um histograma contendo as informações das ocorrências das interpretações nos arquivo da nova base de dados.

4.8 Módulo Analisa

Este módulo é utilizado para a obtenção de padrões numéricos ou simbólicos. Busca-se nos arquivos da BD os padrões pré-definidos segundo uma sintaxe que é especificada abaixo. Os dados encontrados na BD são armazenados em um arquivo no formato planilha do Kards .

Os padrões procurados são codificados em linhas de um arquivo e devem obedecer a seguinte sintaxe, com as seguintes informações:

Termo chave±Palavra auxiliar[21]RestriçãoœTipo do operador¦

onde:

Termo Chave: é a palavra cuja presença dá início a verificação da existência do padrão;

Palavra auxiliar: é a palavra que pode, quando necessário, auxiliar na localização dos padrões;

Restrição: é a palavra que quando presente exclui a obtenção do padrão;

Tipo do operador: os operadores definem a posição do padrão desejado (representada por x) em relação ao termo chave (representado por f). Os operadores criados são os seguintes:

fx - busca a ocorrência posfixa de x;

xf - busca a ocorrência prefixa de x;

f?x - busca a ocorrência posfixa de x, ignorando (?) o que ocorre entre eles;

x?f - busca a ocorrência prefixa de x, ignorando (?) o que ocorre entre eles;

fxg - busca a ocorrência posfixa de x que precede a palavra auxiliar (g);

f?xg - busca a ocorrência posfixa de x que precede a palavra auxiliar (g) não importando as informações prefixas;

fx?? - busca a ocorrência posfixa de x, não importando com a posição da presença da palavra auxiliar;

xf?? - busca a ocorrência prefixa de x, não importando a localização da palavra auxiliar.

As frases onde os padrões foram encontrados são inseridas em arquivos e podem ser utilizadas para reavaliação das sintaxes adotadas. A interface deste módulo é apresentada na figura 4.16.

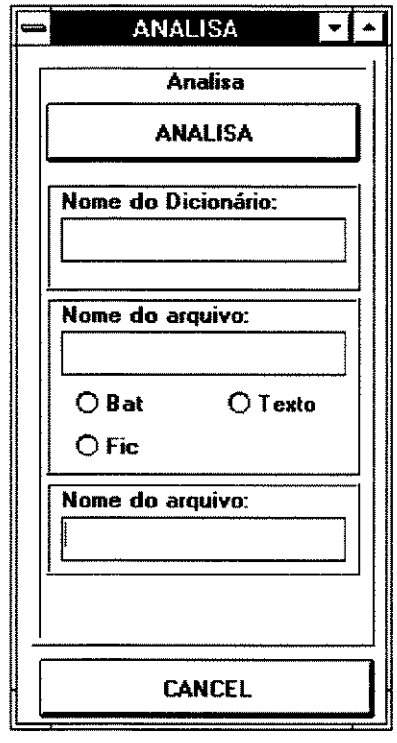

Figura 4.16: Módulo Analisa.

4.9 Ferramentas Auxiliares

As ferramentas auxiliares são operações que auxiliam na utilização dos módulos. Os ícones que ativam as ferramentas são mostrados a seguir.

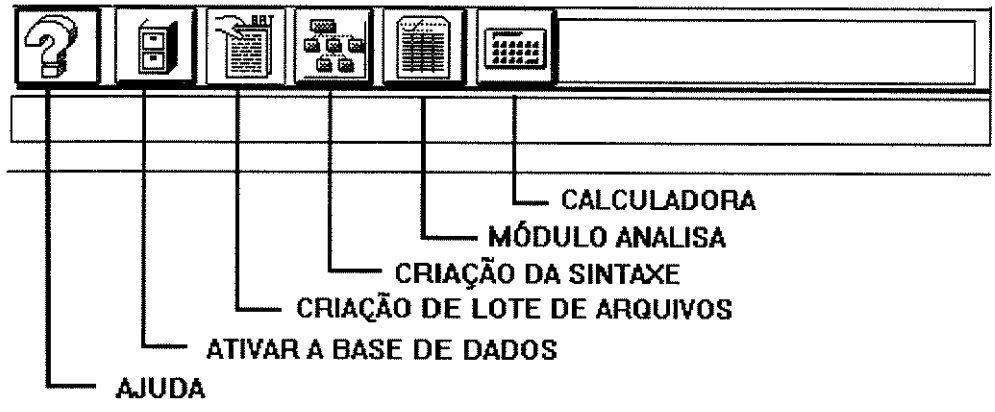

Figura 4.17: Ícones associados às ferramentas.

4.9.1 Criação da Sintaxe

As sintaxes criadas para serem utilizadas no sistema devem ser inseridas ou manuseadas utilizando-se a interface mostrada na figura 4.18. A interface permite operar sobre a área de memória que armazena a sintaxe corrente utilizada no sistema. Assim, ao ativar a interface, a sintaxe corrente é apresentada. Nesta interface, as seguintes ações podem ser executadas sobre a sintaxe corrente: alterar, remover ou inserir símbolos. Novas sintaxes (Define) podem ser criadas e os símbolos correspondentes são em seguida inseridos (Insere).

As sintaxes criadas podem ser guardadas (Salva) em arquivos para serem reutilizadas futuramente (Lê).

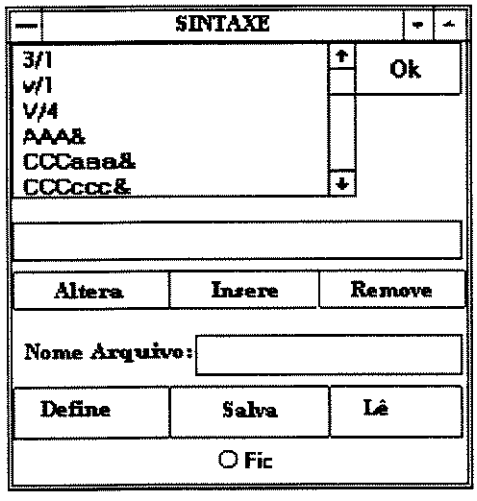

Figura 4.18: Interface Sintaxe.

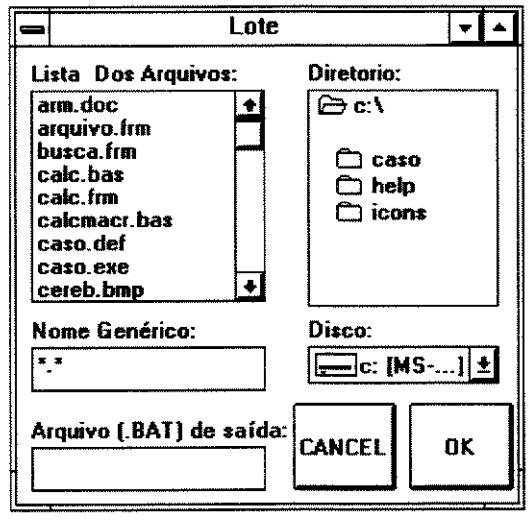

Figura 4.19: Interface Lote.

4.9.2 Criação do Arquivo Lote

Permite a criação de um arquivo contendo os arquivos selecionados na área Lista dos Arquivos da interface apresentada na figura 4.19.

4.9.3 Ajuda

As informações de como utilizar o sistema Jargão podem ser obtidas através da ativação do ícone Ajuda. Estas informações são relativas à utilização dos módulos e das ferramentas, e estão armazenadas na estrutura da base de dados do sistema Kards.

4.9.4 Ativação da Base de Dados

As informações que são obtidas durante as fases da análise podem ser inseridas na estrutura BD do sistema Kards. A seleção do ícone associado a este procedimento ativa o sistema Kards, onde estão implementadas as macros que importam dados (dicionário, sintaxe,...) do formato codificado no Jargão para a BD do sistema Kards. Um aspecto importante a destacar é que o sistema Jargão é um dos subsistemas do sistema Kards.

A estrutura de arquivos da BD do sistema Kards é a seguinte:

a) as informações são instâncias de Fichas que podem ser codificadas na forma de textos, planilhas ou registros;

b) as Fichas são organizadas em Pasta;

c) as Pastas são armazenadas em Gavetas;

d) e as Gavetas são armazenadas no Armário.

Portanto, ao iniciar a utilização do sistema Jargão, ou até antes de ser iniciada a inserção das informações na BD, é necessário especificar no sistema Kards a estrutura da BD. A seqüência a ser seguida é a seguinte:

a) o nome e a estrutura do Armário;

b) o nome e a estrutura das Gavetas;

c) o nome das Pastas e as Fichas que compõe as Pastas;

d) a estrutura de cada Ficha.

Com o objetivo de facilitar a especificação da estrutura da BD, uma estrutura padrão de fichas e pasta é fornecida e pode ser utilizada. Quando adotada a estrutura padrão, torna-se somente necessário a especificação do Armário, Gaveta e da Pasta. Na figura 4.20 mostra-se as fichas que compõem a pasta padrão.

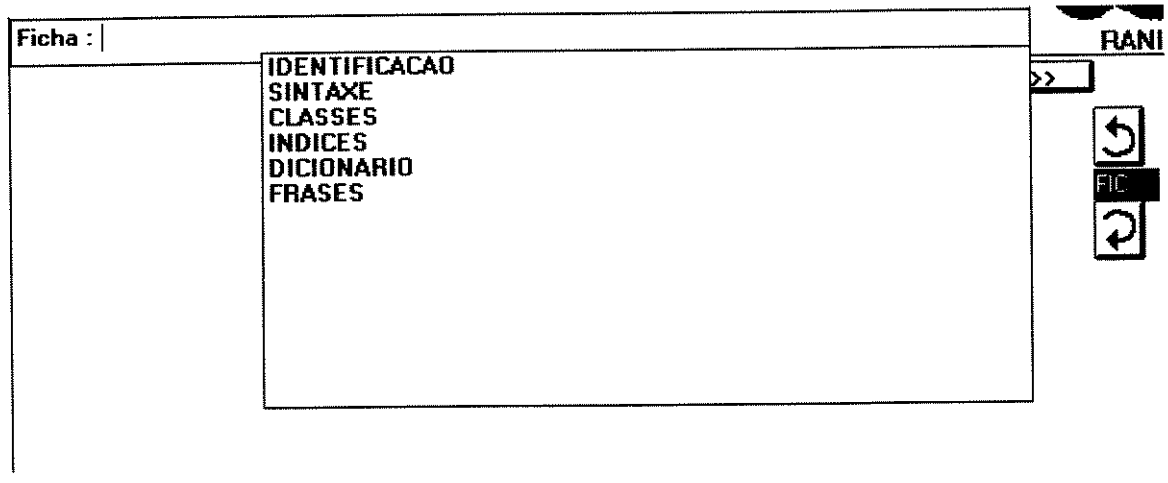

Figura 4.20: Conteúdo da Pasta.

Para cada ciclo de análise executada, uma pasta é criada. As fichas contêm as informações geradas. A ficha de Identificação tem uma estrutura para a descrição simplificada do contexto em que a análise está sendo feita. Uma instância desta ficha é mostrada na figura 4.21.

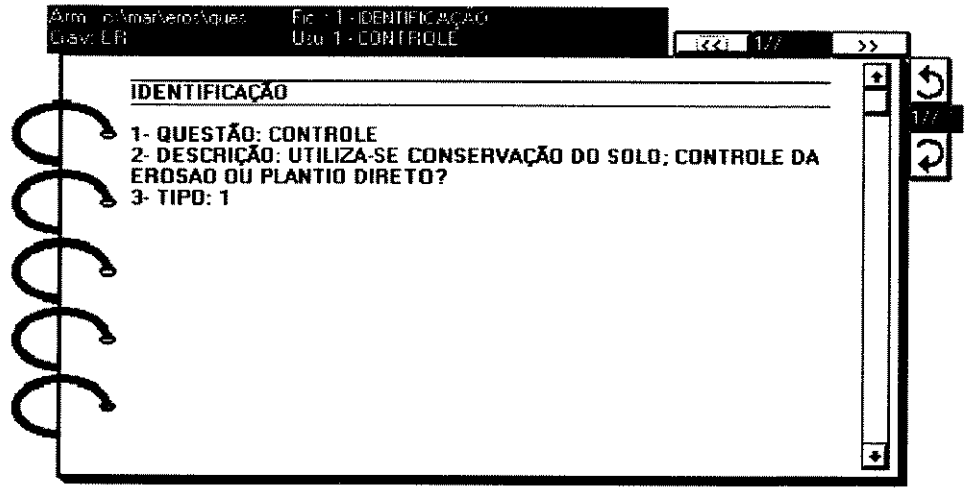

Figura 4.21: Ficha Identificação.

A ficha de sintaxe tem uma estrutura para armazenar a sintaxe criada durante a análise. Uma instância desta ficha é mostrada na figura 4.22.

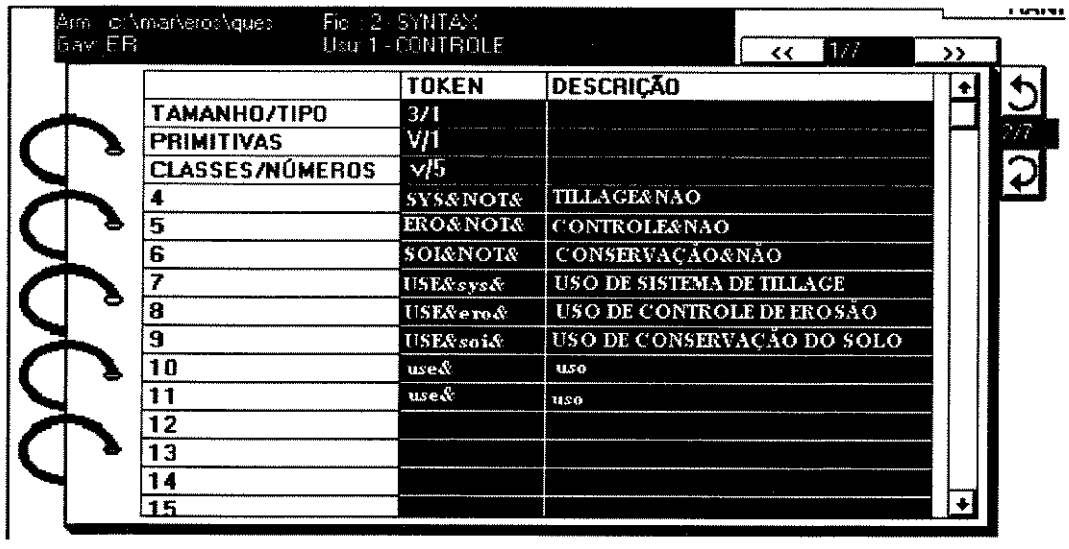

Figura 4.22: Ficha Sintaxe.

A ficha de Classes tem uma estrutura para armazenar o conteúdo do Dicionário de Classes Complexas criado durante a análise. Uma instância desta ficha é mostrada na figura 4.23.

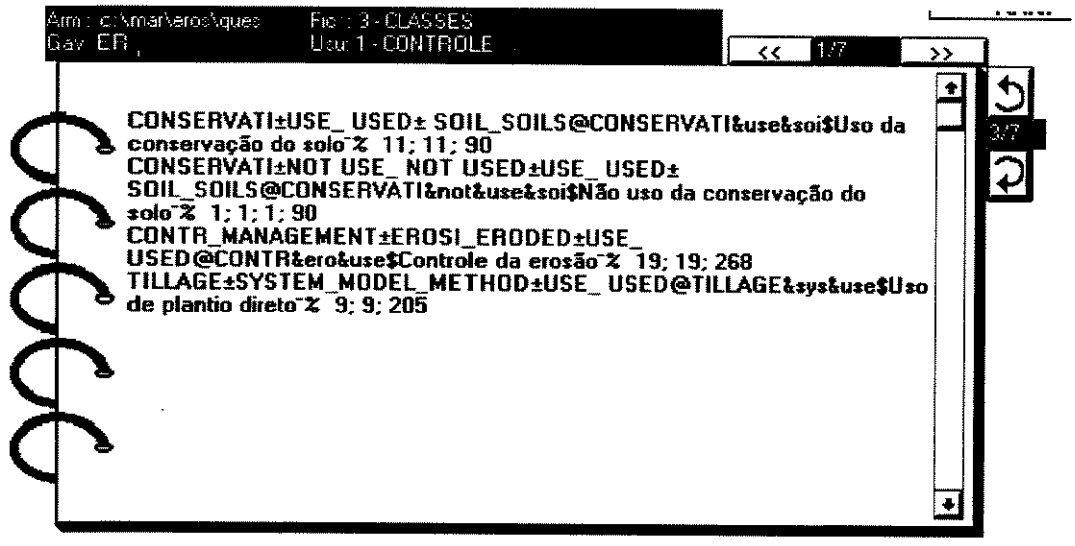

Figura 4.23: Ficha Classes.

A ficha de Índice tem uma estrutura para armazenar os índices da BD onde estão as frases que suportam a definição dos elementos do Dicionário de Classes Complexas. Uma instância desta ficha é mostrada na figura 4.24.

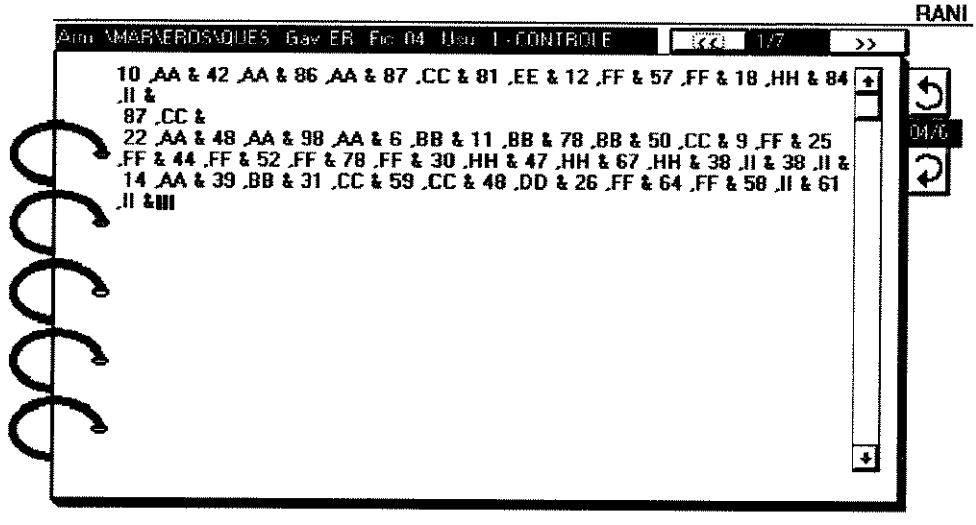

Figura 4.24: Ficha Índices.

A ficha de Dicionário tem uma estrutura para armazenar o conteúdo do Dicionário de Conceitos utilizado na análise. Uma instância desta ficha é mostrada na figura 4.25.

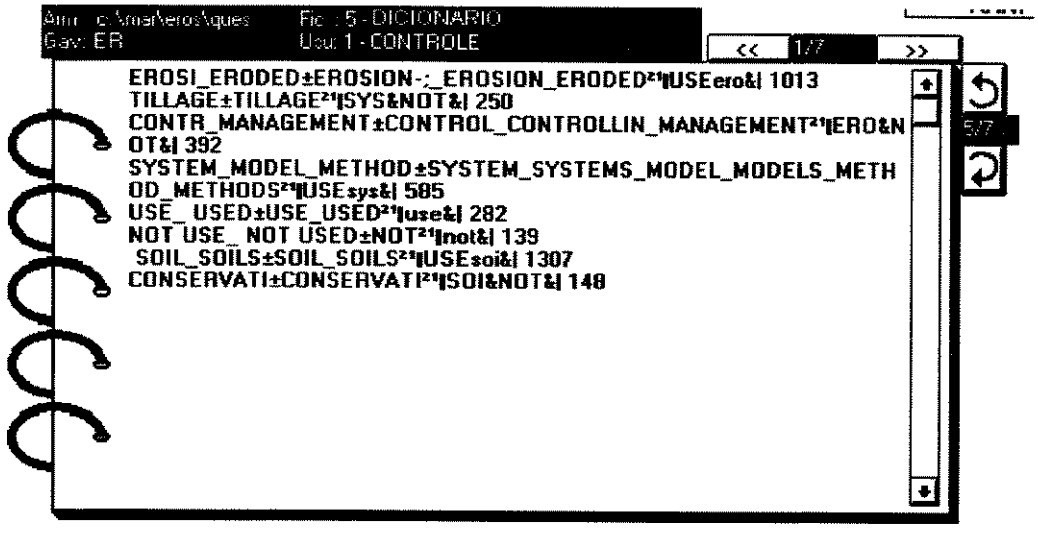

Figura 4.25: Ficha Dicionário.

A ficha de Frases tem uma estrutura para armazenar o conteúdo do Dicionário de Frases criado durante a análise. Uma instância desta ficha é mostrada na figura 4.26.

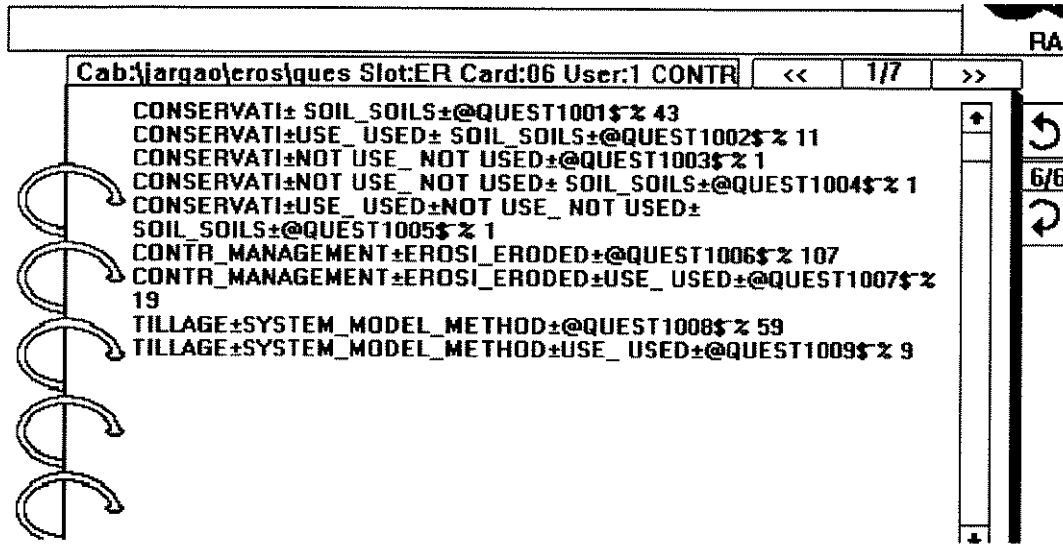

Figura 4.26: Ficha Frases.

4.10 Utilização do Sistema

Na geração de uma RNE e na análise do seu conteúdo são executados, seqüencialmente, os seguintes módulos:

a) Módulo Palavras;

b) Módulo Manuseio;

c) Módulo Combina;

d) Módulo Frases;

e) Módulo Recodifica.

Esta seqüência de ativação dos módulos é utilizada na geração e análise de qualquer um dos níveis hierárquicos de uma RNE hierárquicamente organizada. Assim, a seqüência de ativação dos módulos usados na geração da RNE do Nível de frases são os mesmos utilizados no Nível de Teoria. Portanto, o número de vezes que esta seqüência de ativação dos módulos é executada, equivale ao número de níveis hierárquicos alcançado pela rede gerada.

CAPÍTULO 5

APLICAÇÕES

5.1 Introdução

Neste capítulo, abordam-se algumas das aplicações desenvolvidas utilizando-se o sistema Jargão. Os problemas tratados nessas aplicações são caracterizados por terem uma base de dados em linguagem natural (BDLN) contendo um grande volume de textos (de 1000 a 5000 textos - cada texto contém em média 15 linhas) e que há necessidade de organizar, estruturar e avaliar o conhecimento nela contida, e cujo objetivo é utilizá-lo para automatizar ou prognosticar os processos contidos na BDLN, ou ainda, indexar conceitualmente outras BDLN.

Na primeira utilização apresentada discute-se a utilização do sistema Jargão como uma ferramenta de aquisição de conhecimento em uma BDLN contendo um grande volume de textos (relatórios técnicos ou entrevistas). O conhecimento obtido é utilizado no desenvolvimento de um sistema baseado em conhecimento. Nesta seção são apresentadas as linhas gerais dos passos de um metodologia de aquisição de conhecimento utilizando o sistema Jargão. Esta metodologia resulta das várias análises de BDLN utilizando o sistema Jargão. A descrição dos resultados produzidos na utilização do Jargão como ferramenta de aquisição e a sua utilização em sistemas baseados em conhecimentos são amplamente discutidos e apresentados nos seguintes trabalhos [MIUR91 et alli], [PATR92], [ROCH92b et alli], [SATO92] e [GUIL94].

A seguir, é apresentada uma aplicação desenvolvida para permitir a recuperação conceitual das informações de uma BDLN. Nesta aplicação, descreve-se a utilização do sistema Jargão para construir as representações conceituais das consultas à BDLN. A criação de uma representação conceitual, associada a uma consulta, e as informações recuperadas de uma BDLN a partir dela são apresentadas.

Finalmente, apresenta-se uma aplicação em uma área recente da computação: a Filtragem de Informação. Esta área é semelhante a área de Recuperação de Informação. O sistema Jargão é utilizado na construção das representações conceituais correspondentes às áreas de especializações ou de interesse de um grupo de usuário.

Essas representações são utilizadas para filtrar os novos textos recebidos, sendo somente selecionados para a leitura aqueles relacionados às áreas de interesse. A implementação do processo de filtragem das informações contidas nos textos é construída no sistema Kards.

5.2 Aquisição do Conhecimento em BDLN

A aquisição de conhecimento (AC) é uma das principais atividades do processo de desenvolvimento de sistemas baseados em conhecimento. Existem várias técnicas de aquisição de conhecimento, e elas são dependentes da fonte desse conhecimento. A aquisição de conhecimento através de entrevista é a técnica usada quando a fonte de conhecimento é um ou mais especialistas da área em que se pretende desenvolver o sistema. A entrevista pode ser feita de forma manual ou automaticamente através de programas de computador.

A aquisição de conhecimento de textos (textos técnicos, relatórios ou entrevistas técnicas) contidos em uma BDLN é geralmente feita manualmente. A complexidade da aquisição manual depende da quantidade de textos e também do tamanho dos textos contidos na BDLN. Assim, à medida que aumenta o volume dos textos na BLDN, esta tarefa tem sua complexidade aumentada.

O desenvolvimento de programas visando auxiliar a aquisição de conhecimento de BDLN com grande volume de textos é uma das principais linhas de pesquisa na área de Inteligência Artificial ([McGR89]).

A tarefa de desenvolvimento de sistemas baseados em conhecimento é normalmente executada por um profissional, denominado engenheiro de conhecimento (EC), que é treinado para a aplicação dos seguintes estágios da aquisição do conhecimento: identificação, conceituação, formalização, codificação e teste do sistema, como mostrado na figura 5.1. É importante destacar que o sistema Jargão é utilizado nos estágios de identificação, conceituação e formalização do conhecimento. As fases seguintes de codificação e teste dependem da utilização pretendida do conhecimento obtido e não são executadas utilizando o sistema Jargão.

Na utilização do sistema Jargão como ferramenta de AC, a função de engenheiro do conhecimento é alterada, uma vez que o próprio especialista é quem trabalha sobre a BDLN para produzir o conhecimento. Ao engenheiro do conhecimento cabe a função de orientar o especialista na utilização da ferramenta e dirigir os passos a serem seguidos durante o desenvolvimento do sistema baseado em conhecimento.

A noção que é aceita da função do EC é a sua participação efetiva nas várias fases da AC. Portanto, o EC deve ser treinado na metodologia de AC, ter noções de

Inteligência Artificial e ter fluência e capacidade de organizar idéias. O EC é, geralmente, um profissional capacitado para a tarefa.

Uma outra vertente consiste em contratar um profissional com experiência na área onde pretende-se desenvolver o sistema baseado em conhecimento para desempenhar a função de EC. A principal crítica que se faz a este procedimento é que, ao atuar como EC, o especialista pode codificar o conhecimento de forma tendenciosa ([McGR89]). Deve-se também destacar que o custo de treinamento do especialista como EC é alto, e nem sempre após o treinamento o especialista está capacitado, e também é importante salientar que certas áreas de especialização requerem o desenvolvimento de sistemas baseados em conhecimento justamente por falta de profissionais.

A solução adotada na utilização do sistema Jargão como ferramenta de AC consiste do EC atuar em conjunto com o especialista que pretende desenvolver o sistema baseado em conhecimento, sendo entretanto, que todos os passos são executados exclusivamente pelo especialista. Neste caso, o EC passa a ser o responsável por capacitar e orientar o especialista na utilização da metodologia durante o desenvolvimento do sistema. A vantagem desta abordagem é que durante o desenvolvimento do sistema, o especialista vai sendo capacitado, e depois pode continuar ajustando o conhecimento do sistema desenvolvido ou, até mesmo, atuar na aquisição de conhecimento para o desenvolvimento de outros sistemas baseados em conhecimento.

5.2.1 Estágios da Aquisição do Conhecimento

As técnicas de aquisição são tarefas complexas e consistem freqüentemente na execução de uma seqüência de estágios. Os estágios, segundo Buchanan ([BUCH83 et alli]), são mostrados na figura 5.1.

Os estágios citados na figura 5.1 podem ser aplicados na aquisição do conhecimento contido em BDLN. O estágio da identificação das características consiste na caracterização dos aspectos relacionados com os problemas contidos na BDLN, incluindo os "participantes", a caracterização das fontes e objetivos, e principalmente, os domínios a serem trabalhados.

O estágio da conceituação envolve a especificação dos conceitos primitivos e os conceitos complexos, e as relações entre esses conceitos no domínio. No estágio de formalização, os conceitos, relações e as outras informações obtidas são utilizadas na especificação das subtarefas, dos modelos e dos processos contidos na BDLN. Neste estágio é definida a estrutura formal de representação do conhecimento a ser adotada.

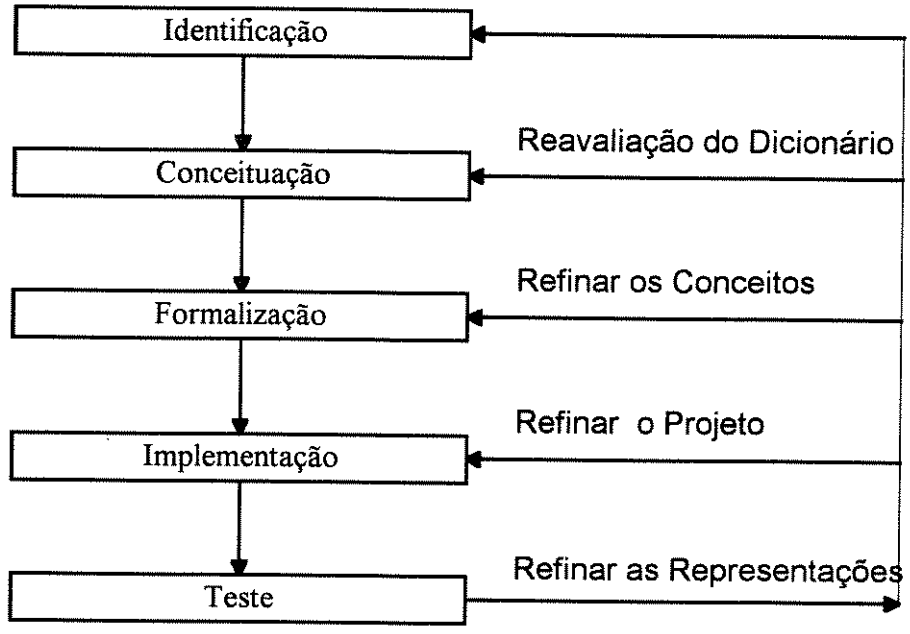

Figura 5.1: Estágios na Aquisição do Conhecimento

O estágio da implementação consiste em inserir o conhecimento formalizado no estágio anterior em um ambiente de desenvolvimento de sistema especialista. A escolha do ambiente deve, evidentemente, levar em consideração a existência do formalismo de representação de conhecimento adotado. Ao final deste estágio tem-se o protótipo do sistema. O estágio de teste requer que o sistema tenha seu desempenho analisado. O procedimento comumente executado consiste em:

a) escolher um conjunto de novos problemas que tenham soluções conhecidas;

b) fornecer os problemas para o sistema e comparar os resultados com as soluções originais.

A análise dos resultado obtidos na comparação podem ser utilizados para verificar a eficácia da aquisição do conhecimento.

Na figura 5.1 pode-se verificar que a execução do estágios são dependentes. Isto significa que a incorreta execução em um estágio pode causar problemas nos estágios seguintes. Portanto, cada estágio deve ser trabalhado exaustivamente, evitando, desta forma, a necessidade de ser refeito futuramente. O sistema Jargão tem sido utilizado ([PATR92] e [SATO92]) como uma ferramenta de apoio à aquisição de conhecimento de BDLN, sendo, como já mencionado, utilizados somente nos três primeiros estágios (Identificação, Conceituação e Formalização).

Nas seções a seguir discute-se detalhadamente a utilização do sistema Jargão nestes três estágios da aquisição de conhecimento de BDLN.

5.2.2 Identificação das Características dos Problemas

5.2.2.1 Visualização Inicial da BDLN

A visualização inicial do conteúdo das BDLN é importante, pois é definido se a mesma pode ser utilizada para o propósito de aquisição de conhecimento. A primeira análise a ser feita, consiste em verificar com os responsáveis pela BDLN, se a mesma contém informações referentes ao domínio do problema ao qual se pretende efetuar a aquisição do conhecimento e se essas informações referem-se às descrições dos procedimentos adotados para solucionar os problemas daquele domínio. Assim, se o domínio que se pretende trabalhar é de uma especialização, por exemplo, a área médica de nefrologia, a área perfuração em petróleo, a BDLN deve conter textos referentes aos procedimento adotados na solução dos problemas da área de especialização, como por exemplo, prontuários médicos de pacientes da área de nefrologia ou relatórios técnicos de perfurações petrolíferas.

Em algumas situações a compreensão da estrutura e as formas de indexação da BDLN podem ser utilizadas para filtrar os textos que devam ser utilizados na análise, e desta forma reduzir o número de textos a serem analisados.

O grau de dificuldade da aquisição pode depender também da qualidade da informação contida na BDLN. Quanto mais geral é o texto, maior é a dificuldade da aquisição do conhecimento. Por outro lado, quanto mais restrito for o domínio, menor é a dificuldade.

A organização da estrutura dos textos pode variar em cada área de especialização. Os textos contidos em uma BDLN de uma especialização geralmente obedecem a mesma estrutura e refletem, em parte, a seqüência dos procedimentos. Assim, a compreensão da organização dos textos fornece uma importante heurística no processo de aquisição. A compreensão permite também descobrir nos textos, a existência de partes contendo informações específicas, como por exemplo, nos prontuários médicos existe uma parte onde estão as informações fornecidas durante a anamnésia, e em outra, o diagnóstico ou a terapia adotada ([SAGE87]). Nestes caso, direciona-se a análise para a parte do texto onde esteja o conhecimento desejado. A adoção desta estratégia pode simplificar em muito, o trabalho de aquisição.

A verificação da forma de segmentação das frases é outro aspecto de grande importância na AC da BDLN. Normalmente, nos textos a segmentação das frases é feita por ponto (.). Podem existir BDLN que utilizam outros separadores ou até mesmo, casos de BDLN onde as frases são pobremente segmentadas. Os textos de frases pobremente segmentadas são mais dificeis de serem analisados.

5.2.2.2 Descrição Inicial da BDLN

Outro aspecto fundamental para a aplicação da metodologia consiste na visualização por parte do engenheiro de conhecimento, do escopo do problema. Isto pode ser feito antes de iniciar-se a utilização do sistema, através de uma entrevista aberta, onde o especialista descreve o ambiente, os processos, os equipamentos utilizados, as pessoas envolvidas, etc.

O resultado da entrevista pode ser inserido na base de dados que está associada ao sistema Jargão. As entrevistas, se desejado, podem ser inseridas em uma base de dados no formato de textos ou através da especificação de campos referentes a: equipamentos, profissionais, diagnósticos, etc. Esta base de dados é uma descrição inicial do ambiente trabalhado e a medida que evolui a análise, seu conteúdo pode ser reavaliado ou também serem inseridas novas informações, produzindo no final, uma base de dados que descreve a BDLN segundo a ótica do especialista. Deve-se destacar que a metodologia fornece uma estrutura inicial para esta base de dados que pode ser redefinida, se desejada, pelo especialista.

5.2.3 Conceituação

A conceituação do domínio é a tarefa mais difícil e também a mais importante do processo de aquisição do conhecimento. A conceituação consiste da abstração do domínio com o objetivo de descobrir os conceitos primitivos, os atributos e valores, e as ações básicas.

Os benefícios que podem advir de uma conceituação adequada são ([McGR89]):

a) a compreensão e a delimitação do domínio;

b) o planejamento e a complexidade da aquisição de conhecimento;

c) as informações desta fase possibilitam uma noção preliminar das características do conhecimento a ser obtido.

Os passos da conceituação utilizando o sistema Jargão são mostrados na figura 5.2. O primeiro passo consiste na criação de um dicionário de conceitos primitivos. A seguir, cria-se uma sintaxe que codifique a representação da estrutura dos conceitos complexos, as relações entre os conceitos e as ações básicas. Os conceitos complexos, que consistem das relações geradas e das ações básicas, são então analisados pelo especialista, que pode verificar a necessidade de reavaliações no dicionário de conceitos ou na sintaxe adotada. A seguir, discute-se cada um dos passos mostrados na figura 5.2.

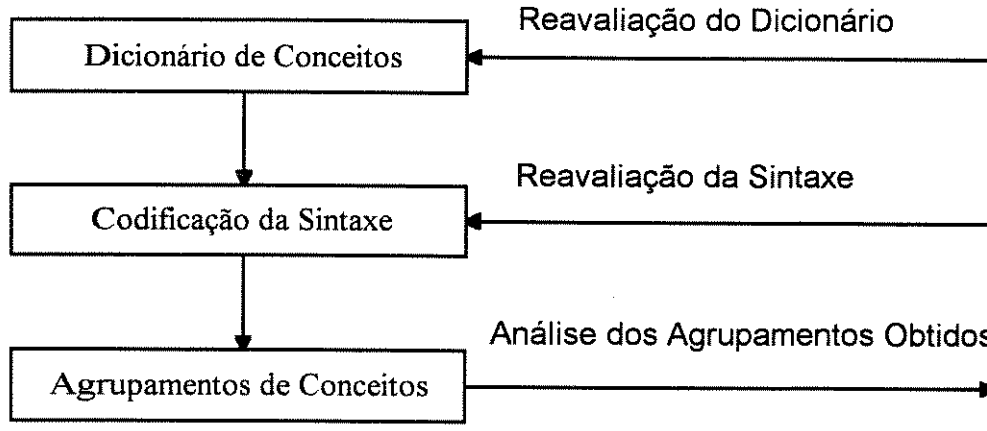

Figura 5.2: Passos da Conceituação.

5.2.3.1 Criação do Dicionário de Conceitos Primitivos

A criação do Dicionário de Conceitos Primitivos é o primeiro passo do processo de conceituação. A criação do Dicionário de Conceitos Primitivos utilizando o Jargão é mostrada nas seções 3.4 e 3.5. O Dicionário de Conceitos contém as palavras encontradas na BDLN com suas respectivas freqüências. Como mostrado na seção 3.5, o especialista analisa as palavras deste dicionário, podendo então:

a) eliminar as palavras que não estão associadas a conceitos ou atributos relacionados ao domínio do problema;

b) agrupar palavras que referem-se a um mesmo conceito;

c) agrupar palavras que referem-se ao mesmo atributo.

Após esta análise o Dicionário de Conceitos Primitivos deve conter somente conceitos primários e atributos relacionados com o domínio em questão.

5.2.3.2 Codificação da Sintaxe

Uma das formas de conceituação é a separação das palavras do dicionário de conceitos nas várias classes de compreensão existentes no domínio trabalhado. O agrupamento destas classes, em certa situações, consiste das estruturas de representação do conhecimento a ser adquirido. Assim, pode-se definir uma sintaxe que permita interpretar e incorporar as estruturas do conhecimento e que permita a representação dos conceitos complexos contidos nos textos. Evidentemente, as interpretações são as instâncias destas estruturas.

A complexidade da sintaxe adotada varia de acordo com o conhecimento conceitual do contexto, podendo variar de uma sintaxe simples a uma complexa. Nas seções 3.6.3 e 3.6.4 são descritos os passos para a definição de uma sintaxe.

A codificação sintática consiste em definir o conjunto de símbolos associados às classes, correspondendo aos transmissores, receptores e controladores que direcionam a criação da RNE. O nível de conhecimentos a ser gerado é equivalente a geração da Rede de Conceitos Complexos mostrados na seção 3.6.

A definição das classes e a codificação sintática nos dicionários de conceitos primitivos com poucos termos é geralmente uma tarefa simples e é feita em um único passo. A medida em que se aumenta o número de termos no dicionário e dos símbolos das classes sintáticas, este procedimento torna-se um pouco complexo podendo ser efetuado em partes. Assim, uma das seguintes estratégias pode ser utilizada:

a) define-se primeiramente a sintaxe relacionada com as estruturas de conhecimento referentes ao paradigma declarativo e em seguida, incrementa-se a mesma sintaxe para operar também com o paradigma procedimental; ou,

b) define-se inicialmente uma sintaxe simples e incrementa-se a sua complexidade à medida que torne-se necessária.

Portanto, após ter sido definida a sintaxe, deve-se então atribuir aos termos do dicionário de conceitos as diferentes classes correspondentes.

5.2.3.3 Geração dos Agrupamentos

Os agrupamentos gerados neste nível correspondem aos módulos das Redes Conceitos Complexos, cuja estrutura é mostrada na seção 3.6. Assim, o algoritmo de geração dos agrupamentos implementado é feito de acordo com:

a) a sintaxe codificada; e

b) a atribuição, pelo usuário, dos símbolos da sintaxe aos elementos do dicionário de conceitos primitivos; e

c) a existência dos agrupamentos nos textos.

A geração de um agrupamento é feita quando uma palavra classificada na classe definida pelo especialista como primária é encontrada na BDLN. A classe primária pode então, requerer ou não, o encadeamento de outras classes não primárias. Se as palavras associadas a estas classes são encontradas, temos um agrupamento. A classe primária é aquela associada aos verbos que descrevem ações, quando a BDLN contém conhecimento procedimental. Nas BDLN contendo conhecimento declarativo, ela está associada a termos que referem-se explicitamente ao conceito. As classes que não são primárias têm a função de complementar a descrição ou as ações associadas à classe primária. Grande parte da ambigüidade da gramática é eliminada pelos textos contidos na BDLN.

5.2.3.4 Análise dos Agrupamentos

Após gerar os agrupamento, o especialista pode analisar o resultado de duas formas distintas: conhecimento específico ou conceitual. A avaliação de cada uma das formas pode ser vista na seção 3.6.5. Na primeira forma, que é aplicada para os agrupamentos gerados com qualquer nível de conhecimento sintático (simples ou complexos), os agrupamentos consistem de cadeias de palavras ou símbolos criados de acordo com a sintaxe previamente definida. Neste caso, um a um dos agrupamentos ou interpretações (veja a seção 3.6.5) encontradas são analisadas verificando-se, a coerência do conhecimento nele contido com os conceitos complexos existentes no contexto trabalhado. Os agrupamentos consistentes são inseridos no Dicionário de Frases.

A outra forma, é utilizada quando a sintaxe adotada é complexa, os agrupamentos consistem das seqüências de acoplamentos entre as classes sintáticas com as respectivas instâncias (termos do Dicionário de Conceitos Simples). Neste tipo de análise, os agrupamentos correspondem às estruturas de conhecimento com as respectivas instâncias. Nesta análise, o usuário verifica se os agrupamentos gerados correspondem ou não, às estruturas de conhecimento esperadas. Se as estruturas estiverem consistentes com o domínio, significa que as definições anteriores foram feitas corretamente, podendo então ser inseridas no Dicionário de Conceitos Complexos. Caso contrário, deve-se diagnosticar de qual passo anterior da conceituação os problemas decorrem. Quando isto ocorre, deve-se corrigir os problemas e refazer os passos subseqüentes.

Após a análise dos agrupamentos e a verificação de que todos os conceitos necessários foram obtidos, o conhecimento contido no dicionário (Frases e Conceitos Complexos) é utilizado para recodificar a BDLN. Desta forma, é criada uma nova BD recodificada, contendo somente os conceitos definidos.

5.2.4 Formalização do Conhecimento

Após ser feito o processo de conceituação, onde encontrou-se os conceitos e as ações básicas presentes na BDLN, deve-se, nesta fase, enfocar a obtenção e estruturação do conhecimento de mais alto nível, que é associado à especificação das tarefas e problemas. Este fase corresponde ao nível de teoria mostrado na seção 3.9.

Quando o objetivo da AC é a obtenção de conhecimento para o desenvolvimento de um sistema baseado em conhecimento, deve-se paralelamente, iniciar a análise dos problemas do domínio com o objetivo de definir as características do sistema. Esta análise é feita utilizando metodologias de Engenharia de Software.

As metodologias de projetos de sistemas e aquisição do conhecimento geralmente seguem uma abordagem onde os problemas ou sistemas são divididos em pedaços ou tarefas menores, até chegar ao nível de especificação dos conceitos. A metodologia de projetos de sistemas e aquisição de conhecimento utilizada no sistema Jargão segue uma abordagem oposta, um vez que procura-se definir inicialmente as partes ou tarefas básicas, e estas, são agrupadas em estruturas hierárquicas de conhecimento para a representação dos problemas e sistemas presentes no domínio.

Assim, a fase de análise, uma das fases da metodologia de engenharia de software, é iniciada paralelamente a esta fase de aquisição de conhecimento. Essas atividades são desenvolvidas em paralelo, uma vez que, nesta fase, procura-se definir claramente quais são os problemas presentes no domínio e as estratégias existentes para solucioná-los. Portanto, essas informações são de grande importância para o projeto do sistema, isto porque, o sistema deve ser construído de acordo com a dinâmica existente nas várias tarefas presentes no domínio.

O primeiro aspecto a ser considerado na fase de formalização é o tipo de conhecimento presente nos textos que pode ser os seguintes: procedimental, declarativo, ou episódico. Em seguida, o usuário executa os passos mostrados na figura 5.3. No primeiro passo, opera-se com as informações obtidas nas fases anteriores, ou seja, as estruturas, ou as interpretações, ou as ações. Estas informações compõem o que denominamos de dicionário de frases (ou Conceitos Complexos). Assim, para se obter o conhecimento hierarquicamente mais complexo, novamente deve ser definida uma sintaxe que interprete o tipo de conhecimento presente no texto. A partir da sintaxe e do dicionário são encontrados os agrupamentos. Os agrupamentos são analisados, e em função dos resultados pode-se: requerer a reavaliação do dicionário e/ou da sintaxe, ou considerá-lo representativo para o nível de conhecimento desejado. Pode-se verificar que os passos desta fase são basicamente os mesmos aplicados na fase de conceituação. Nas subseções seguintes discute-se os passos.

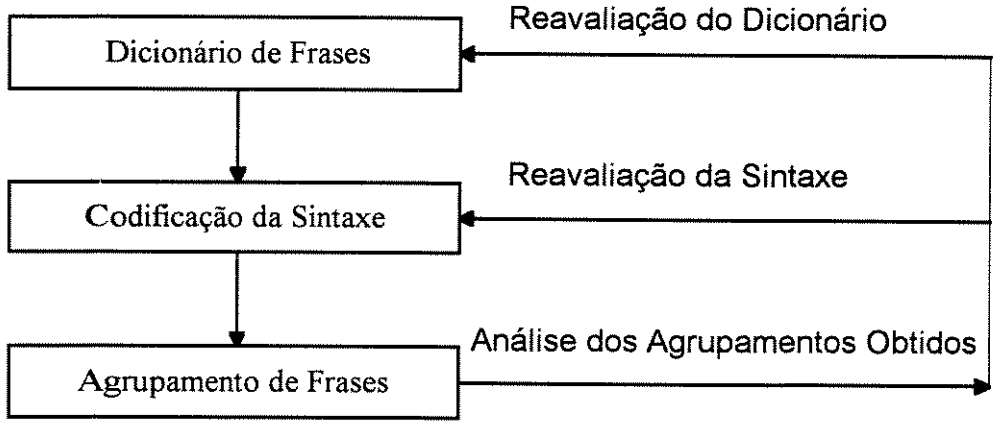

Figura 5.3: Passos da Formalização.

5.2.4.1 Operações sobre o Dicionário de Frases (Conceitos Complexos) e a Definição da Sintaxe

O forma de análise adotada na seção anterior é que determina o dicionário (Frases ou Conceitos Complexos) a ser utilizado. Em seguida, os termos (estruturas ou interpretações) presentes no dicionário que tenham mesmo significado semântico são agrupados em uma única representação.

O próximo passo consiste da definição da sintaxe, que é feita de forma a interpretar a estrutura de conhecimento desejada. Evidentemente, é importante levar em consideração: o tipo do conhecimento (procedimental ou declarativo); e o nível hierárquico de especificação do conhecimento trabalhado. Assim, a definição sintática é feita de modo que:

a) no processamento declarativo agrupe as características associadas à descrição do sistema;

b) no processamento procedimental agrupe as ações associadas às tarefas executadas no domínio.

Quanto ao nível hierárquico, a sintaxe deve ser criada para agrupar conhecimento de uma mesma classe hierárquica, de forma a permitir a descrição, primeiramente das subtarefas, para depois poder obter as tarefas; e das subfunções, para depois poder obter as funções, etc.

Assim, levando-se em consideração os aspectos levantados acima, são especificados os símbolos correspondentes às classes sintáticas. A seguir, eles são atribuídos aos correspondentes termos do dicionário.

5.2.4.2 Geração e Análise dos Agrupamentos

Os agrupamentos gerados nesta fase correspondem aos módulos da Rede de Teorias mostrados na seção 3.9.3., e são criados de acordo com: a sintaxe codificada; a atribuição dos símbolos da sintaxe atribuídos aos elementos do dicionário de frases; a existência dos agrupamentos nos textos recodificados.

A geração de um agrupamento é feita quando uma frase classificada na classe definida pelo especialista como primária é encontrada. A classe primária pode, então, requerer ou não o encadeamento de outras classes não primárias. Se as palavras associadas a essas classes são encontradas, temos um agrupamento.

As classes primárias devem ser associadas às frases que estão explicitamente associadas ao conhecimento a ser obtido, ou seja, são frases que remetem aos problemas. As classes que não são classificadas como primárias têm a função de complementar as descrições ou as ações associadas à classe primária.

Após terem sido gerados os agrupamentos, o especialista deve verificar se o resultado corresponde às estruturas de conhecimento desejadas. Caso não correspondam, o usuário deve, como mostrado na figura 5.3, verificar se os problemas advêm de erros nos passos anteriores. Os possíveis erros devem ser corrigidos e os passos subseqüentes devem ser refeitos.

Quando os agrupamentos gerados são satisfatórios por refletirem o conhecimento desejado, o especialista deve então:

a) verificar se o conhecimento produzido nos agrupamentos correspondem ao nível hierárquico e de especificação do conhecimento desejados, não sendo, portanto, necessária a continuação da análise. Caso contrário;

b) deve-se recodificar novamente a base de dados anteriormente recodificada com os agrupamentos obtidos, e refazer sobre a nova base recodificada todos os passos feitos na fase Formalização.

5.2.5 Discussões

O maior problema que se depara na aquisição de conhecimento de textos técnicos é a pobreza de detalhes e muitas vezes a ausência das informações. Os profissionais de áreas técnicas que redigem os textos geralmente acreditam que certas informações estão subentendidas e não necessitam serem expressas. As informações subentendidas devem então ser fornecidas pelo especialista durante a interação com o sistema. Evidentemente, o conhecimento a ser codificado no processo de aquisição torna-se mais rico que o contido nos textos.

Outro aspecto a ser levantado é que as informações contidas nos textos geralmente não explicitam a dinâmica das ações executadas pelo especialista durante a sua atividade e certamente não permitem a visualização de como o sistema inteligente deve ser especificado.

Portanto, é importante então frisar que o especialista além de ser responsável pelo conhecimento contido no sistema, deve interagir com o E.C. para a definição dos requerimento do sistema inteligente, para que este venha realmente suprir as expectativas dos profissionais que venham utilizá-lo. Por este motivo, durante a fase de formalização o especialista deve também preocupar-se em definir claramente o papel que o sistema deverá desempenhar.

5.3 Recuperação Conceitual das Informações

A área de Recuperação de Informações visa desenvolver mecanismos computacionais que possibilitem ao usuário localizar de forma eficiente alguma informação em base de dados com grande volume de dados ([BELK92]). Conceitos provenientes da Inteligência Artificial estão sendo utilizados para permitir a recuperação conceitual das informações ([MAUL91]). Nesta sessão é apresentada uma aplicação para a recuperação conceitual de informação.

5.3.1 O Domínio do Problema

O domínio a ser trabalhado consiste de aproximadamente 600 resumos de artigos, em inglês, relacionados à Erosão do Solo, filtrados de uma Base de Dados (CAB Abstract on CD-ROM da SilverPlatter Information). Os resumos foram inseridos em uma BD no formato do sistema KARDS.

Normalmente, deseja-se recuperar os resumos relacionados a alguns assuntos específicos. Os assuntos específicos podem ser associados às consultas. Neste contexto, utiliza-se o sistema Jargão para criar as estruturas de conhecimento, denominadas de Representações, referente às consultas. As representações são construídas de maneira a serem utilizadas na recuperação e indexação conceitual dos resumos. Os textos recuperados são analisados, podendo ser aceitos ou requererem a modificação do dicionário e/ou da sintaxe (Figura.5.4).

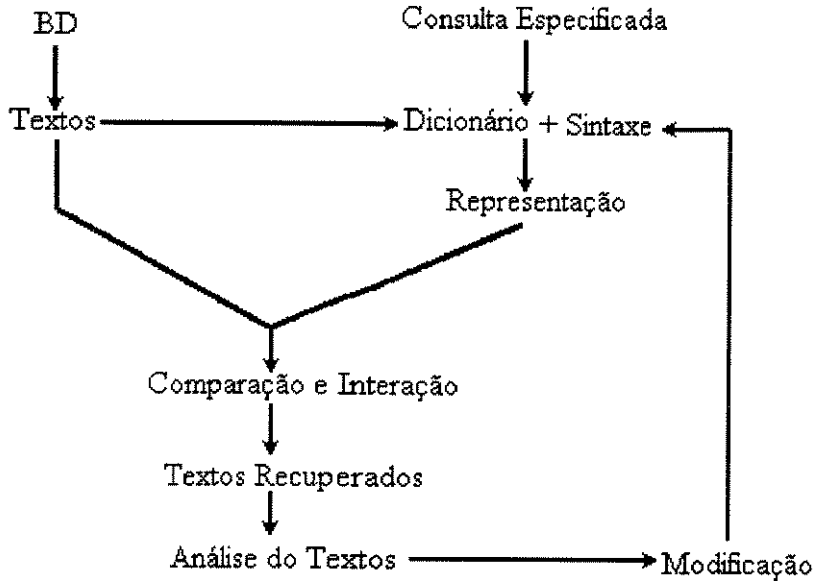

Figura 5.4. Esquema para Recuperação de Informação de Textos Utilizando o Jargão.

5.3.2 Criação do Dicionário

O primeiro passo na definição das estruturas das questões consiste na obtenção do Dicionário de Conceitos Primitivos, que é composto das palavras existentes nos resumos.

A partir do Dicionário de Conceitos Primitivos cria-se um dicionário reduzido contendo apenas termos associados à questão proposta.

Na criação da consulta "A conservação do solo, o controle da erosão e o plantio direto são práticas utilizadas ou não?", as palavras procuradas dentro das frases da BD são: USO (USE), CONSERVAÇÃO (CONSERVATION), SOLO (SOIL), CONTROLE (CONTROL), EROSÃO (EROSION), PLANTIO DIRETO (TILLAGE), SISTEMAS (SYSTEM) e NÃO (NOT). Assim, utilizando os termos do Dicionário de Conceitos Primitivos cria-se um dicionário (QUEST1.DIC) específico contendo apenas os termos referentes às palavras acima (figura 5.5).

5.3.3 Definição da Sintaxe

O passo seguinte consiste da definição de uma sintaxe que especifique os possíveis encadeamentos entre as palavras existentes nessa consulta, representando, conseqüentemente, as relações entre os conceitos. O primeiro passo é verificar entre os termos do dicionário quais são os termos chaves e os complementos associados. A

seguir, define-se uma sintaxe cujas classes sintáticas especifiquem os encadeamentos existentes entre os termos chaves e os complementos. A sintaxe especificada para a questão acima é a seguinte:

SYS&NOT& // Termo Chave 1
ERO&NOT& // Termo Chave 2
SOI&NOT& // Termo Chave 3
USEsys& // SÍMBOLO NÃO TERMINAL associado ao Termo Chave 1
USEero& // SÍMBOLO NÃOTERMINAL associado ao Termo Chave 2
USEsoi& // SÍMBOLO NÃO TERMINAL associado ao Termo Chave 3
use& // SÍMBOLO TERMINAL
not& // SÍMBOLO TERMINAL

Os encadeamentos possíveis para a sintaxe acima são os seguinte.

not& ← SYS&NOT& → USEsys&
↘
not& ← ERO&NOT& → USEero& → use&
↗
not& ← SOI&NOT& → USEsoi&

As classes sintáticas definidas na sintaxe acima são atribuídas aos termos do dicionário, como mostrado na figura 5.5.

QUEST1.DIC:

CONCEITOS PRIMITIVOS	PALAVRAS	FREQ	SINTAXE
EROSION	EROSI_ERODED_EROSION-;_	1013	USEero&
TILLAGE	TILLAGE	250	SYS&NOT&
CONTROL	CONTR_MANAGEMENT_ CONTROLLIN_MANAGEMENT	392	ERO&NOT&
SYSTEM	SYSTEM_MODEL_METHOD_	585	USEsys&
USE	USE_ USED_APPLI_	453	use&
NOT USE	NOT USE_	139	not&
SOIL	SOIL_	1307	USEsoi&
CONSERVATION	CONSERVATI_	148	SOI&NOT&

Figura 5.5 : Dicionário de Conceitos Simples.

Portanto, de acordo com as definições sintáticas e as atribuições das classes sintáticas aos termos do dicionário, temos:

NÃO USO ← PLANTIO DIRETO → SISTEMA
↘
NÃO USO ← CONTROLE→ EROSÃO → USO
↗
NÃO USO ← CONSERVAÇÃO → SOLO

5.3.4 Geração e Análise dos Conceitos Complexos

Uma vez definida a sintaxe e o Dicionário de Conceitos Primitivos, são extraídos da BD todos os Conceitos Complexos (frases) onde encontram-se as interpretações: palavras do dicionários combinadas de acordo com a sintaxe dada. As interpretações encontradas segundo a sintaxe e o Dicionário definidos acima são mostradas a seguir.

```
CONSERVATI± SOIL_SOILS@$¨% 43; 89
CONSERVATI±USE_USED±SOIL_SOILS@$¨% 11;11;899
CONSERVATI±NOT USE_ NOT USED@$¨% 1; 89
CONSERVATI±NOT USE_ NOT USED± SOIL_SOILS @$¨% 1; 1; 89
CONSERVATI±NOT USE_NOT USED±USE_USED±SOIL_SOILS@$ ¨% 1;1;1; 89
CONTR_MANAGEMENT±EROSI_ERODED@$¨% 106; 267
CONTR_MANAGEMENT±EROSI_ERODED±USE_ USED@$¨% 19; 19; 267
TILLAGE±SYSTEM_MODEL_METHOD@$¨% 60; 205
TILLAGE±SYSTEM_MODEL_METHOD±USE_USED@$¨%9;9; 205
```

Nas seções anteriores mostrou-se que as análises dos Conceitos Complexos são feitas de duas formas distintas: específica e conceitual. Nesta aplicação, foi utilizado a análise conceitual discutida nas seções 3.7.2 e 4.6.2.

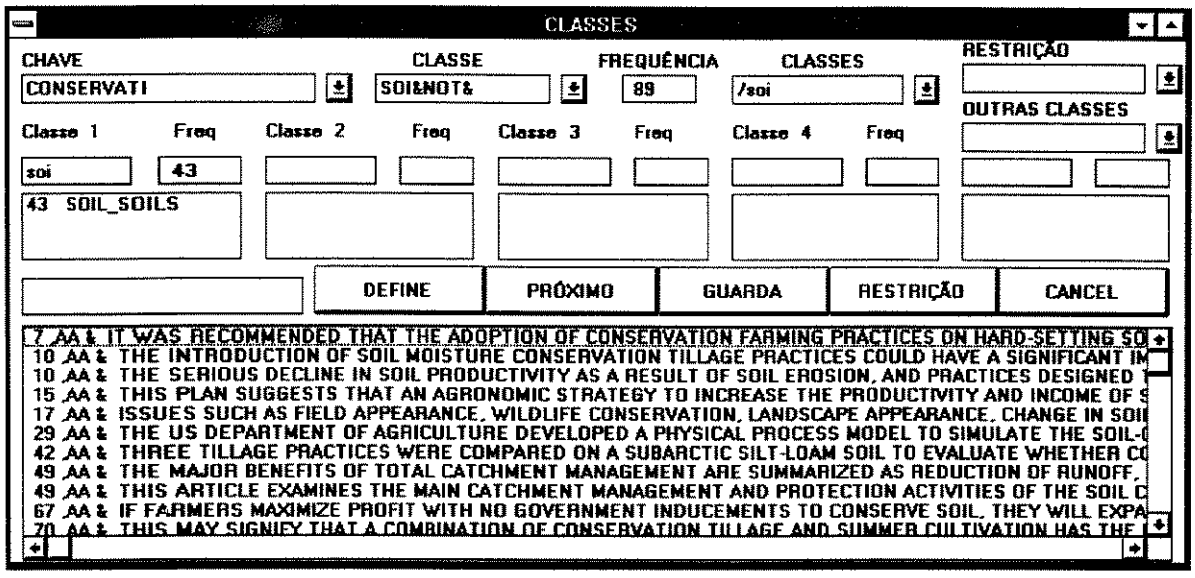

Figura 5.6: Interface mostrando a interpretação "CONSERVATI± SOIL_SOILS".

As interpretações obtidas (mostradas acima) são apresentadas (a interface do sistema referente a esta fase é mostrada na figura 5.6) junto com as frases onde elas ocorrem (figuras 5.6 a 5.9) para que sejam analisadas, podendo:

a) não corresponder às informações desejadas, isto porque as respostas para a consulta não existem na BD;

b) corresponder às informações desejadas, entretanto, sua especificidade pode ser melhorada através de uma reavaliação na sintaxe e/ou no Dicionário de Conceitos Primitivos;

c) inequivocamente corresponder às informações desejadas.

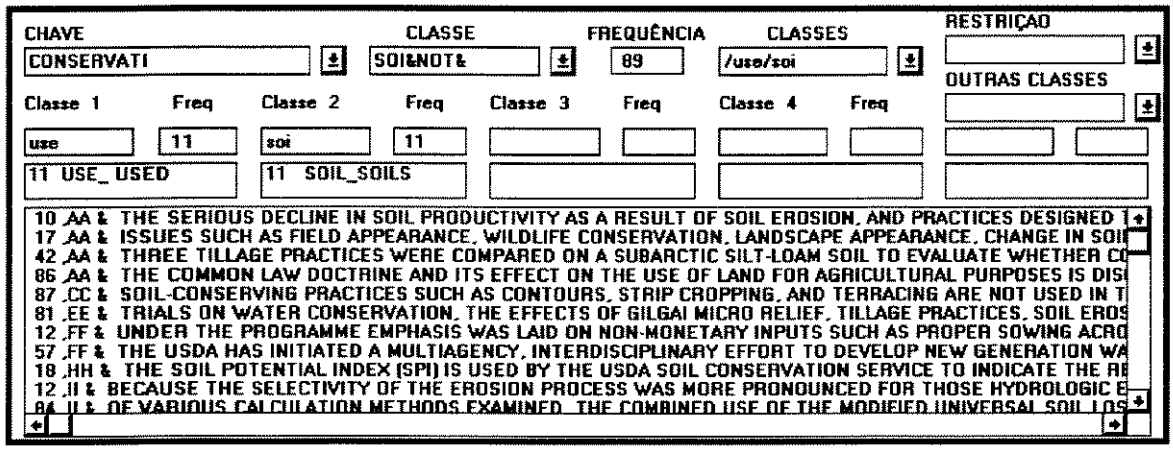

Figura 5.7: Interface mostrando a interpretação "CONSERVATI± USE_USED± SOIL_SOILS".

Na análise das estruturas e das frases no contexto da questão acima verificou-se que as informações desejadas foram recuperadas nas estruturas apresentadas nas figuras 5.7, 5.8, 5.9, 5.10. Essas estruturas compõem o dicionário de conceitos complexos (Figura 5.11). As outras estruturas que são muito genéricas ou que não se referem as informações requisitadas na questão, são descartadas. Isto ocorre, por exemplo, com a estrutura apresentada na figura 5.6., que foi descartada por ser genérica demais em relação à estrutura apresentada na figura 5.7.

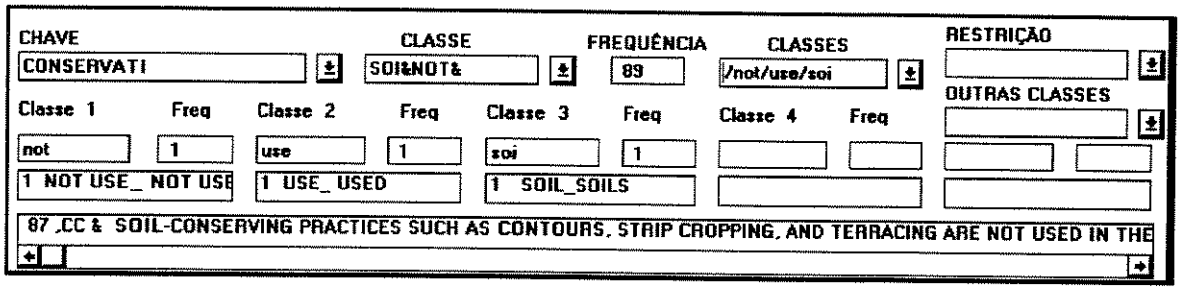

Figura 5.8: Interface mostrando a interpretação "CONSERVATI±NOT USE_NOT USED ±USE_USED±SOIL_SOILS".

Nas situações em que na análise das estruturas apresentadas seja verificado que as mesmas não são suficientemente consistentes, é necessário a reavaliação do dicionário e/ou da sintaxe, ou até mesmo, significa que as informações desejadas não estão presentes na BD.

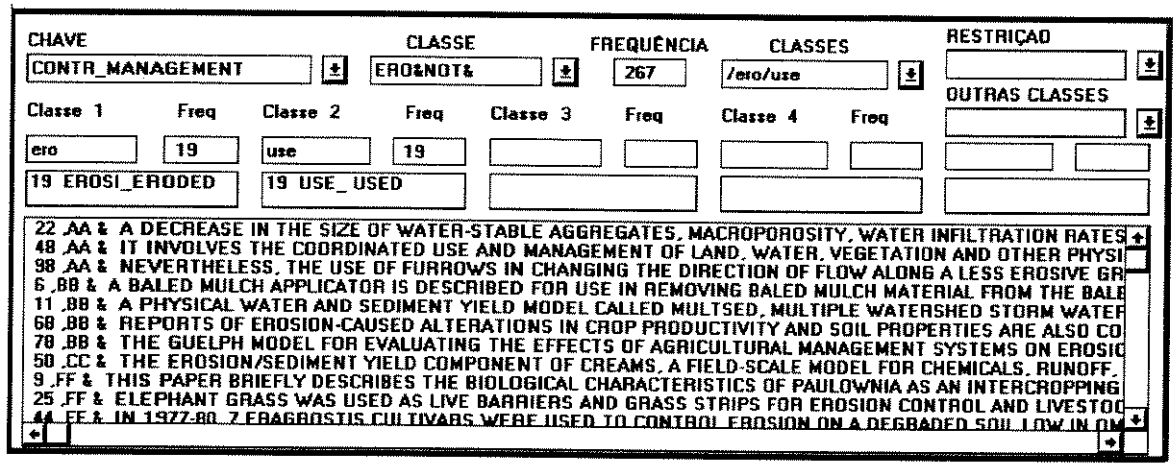

Figura 5.9: Interface mostrando a interpretação "CONTR_MANAGEMENT± EROSI_ERODED±USE_ USED".

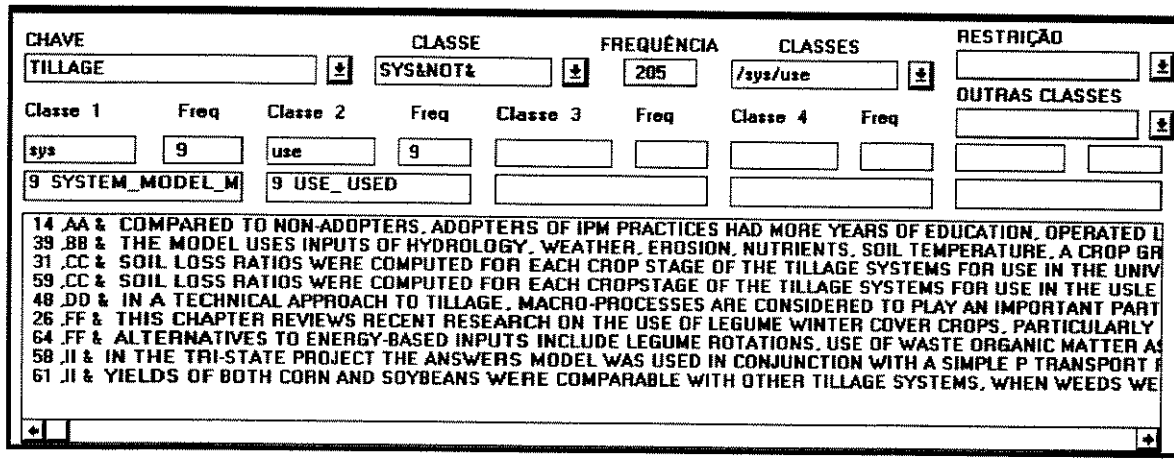

Figura 5.10: Interface mostrando a interpretação "TILLAGE± SYSTEM_MODEL_METHOD± USE_USED".

Os índices dos textos da BD onde estão as frases que contêm as estruturas são também armazenadas, podendo ser utilizados futuramente como referência ou indexação de resumos relativos à questão. Assim, para cada uma das estruturas aceitas, são armazenados os índices das frases onde sua presença foi verificada. Na figura 5.12 mostram-se os índices associados às estruturas aceitas na fase de análise e que referem-se à questão proposta. Um exemplo da utilização dos índices é mostrado na próxima seção.

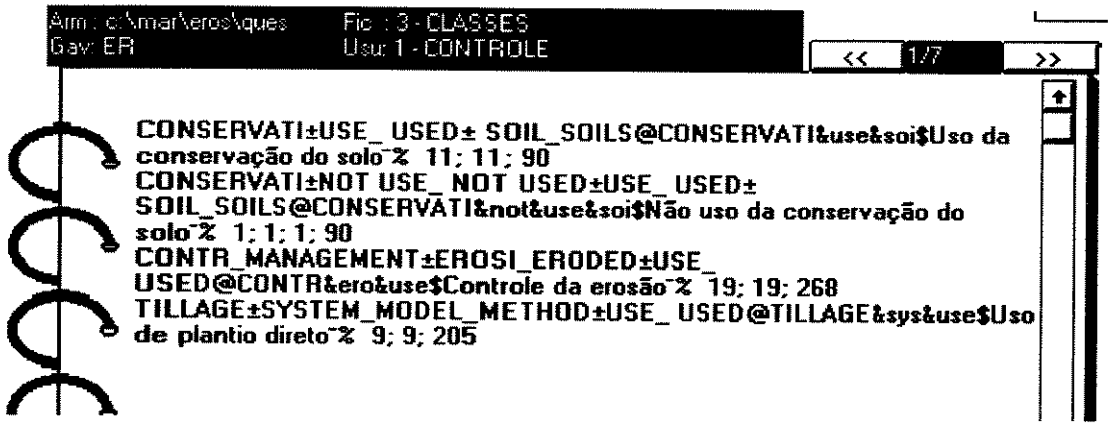

Figura 5.11: Dicionário de Conceitos Complexos.

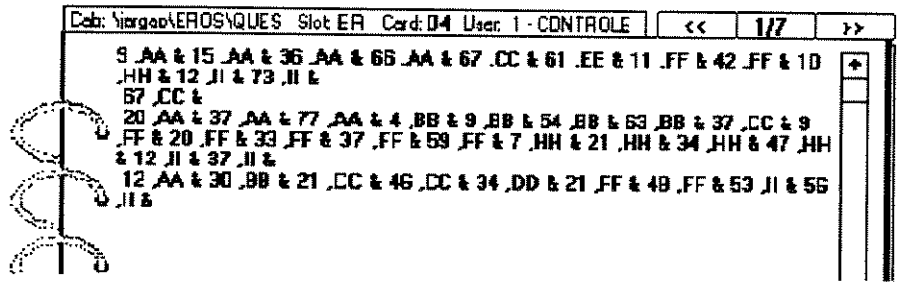

Figura 5.12: Índices dos Textos Associados aos Conceitos Complexos.

5.4 Filtragem de Informação

A filtragem de informação é utilizada nas situações em que pretende-se filtrar de um conjunto de informações recebidas, aquelas informações regulares que estão relacionadas às áreas de interesses de um ou mais usuários que atuam em áreas especificas ([BELK92]).

A dinâmica desta aplicação consiste em verificar nos textos recebidos, a ocorrência das estruturas de conhecimento, denominadas de representações, relacionadas a assuntos específicos e que estão armazenadas na BD de representações. Os textos onde as representações ocorrem são analisados determinando a sua utilização e também direcionando as possíveis modificações no dicionário e/ou na sintaxe (Figura 5.13).

Nos sistemas de filtragem de informações, uma das principais dificuldades consiste na criação das estruturas e no conteúdo das representações. Os problemas decorrem do fato das estruturas das representações serem dependentes das estruturas e do conteúdo dos textos a serem recebidos. Normalmente, as estruturas são codificadas de forma independente dos textos. Para a utilização do Jargão na criação das representações é necessário que exista um conjunto de textos anteriormente recebidos e relacionados com as áreas de interesse. A utilização dos textos recebidos anteriormente possibilitam a geração de representações consistentes com o jargão e com as estruturas dos textos a serem futuramente recebidos.

A geração das representações é feita utilizando o sistema Jargão, e o processo de comparação dos textos recebidos com as representações é feita utilizando os recursos do sistema Kards.

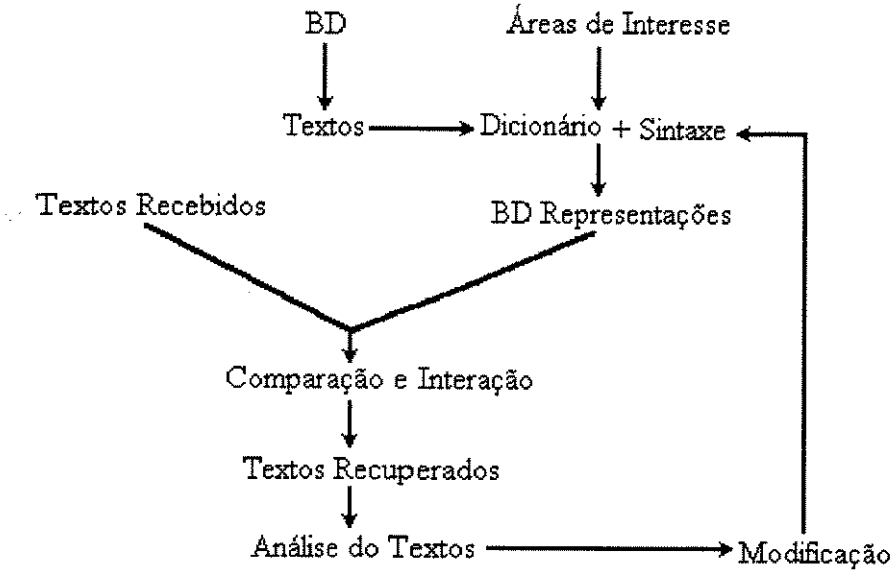

Figura 5.13: Esquema de Filtragem de Informações Utilizando o Jargão.

5.4.1 Criação da BD de Representações

A base de dados utilizada é a mesma da seção anterior. Contém cerca de 600 resumos de artigos relacionados com Erosão do Solo.

A geração das representações é feita utilizando o Jargão, da mesma forma das consultas na seção anterior. Portanto, à partir dos textos da BD é criado um dicionário contendo as palavras neles presentes. Assim, para cada área de interesse, a partir do dicionário cria-se o dicionário específico, e uma sintaxe que codifique as relações existentes entre os elementos do dicionário específico. Em seguida, as representações são criadas de acordo com o dicionário e a sintaxe definida. As representações criadas que não correspondem a conceitos da área em questão são descartadas.

A estrutura da BD de representações é criada utilizando-se o sistema Kards. Portanto, de acordo com as definições mostradas na seção 4.9.4, as informações relativas às áreas de interesse são organizadas em um armário. Para cada área de interesse é criado uma gaveta, como por exemplo, neste contexto, a relacionada com o assunto Erosão do Solo. Nas gavetas são criadas as pastas que correspondem às subdivisões: Controle; Preços; Fertilizantes; Plantio Direto; Remoção; Efeitos; e Redução de Efeitos (figura 5.14). Em cada pasta são inseridas as fichas correspondentes: a identificação; sintaxe; classes; dicionário; frases e os índices.

Como mostrado na seção 4.9.4, as informações geradas no Jargão podem ser exportadas para estruturas criadas no sistema KARDS. Nesta aplicação, as informações

importadas são: o dicionário, a sintaxe, as classes e o índice. O conteúdo das fichas da pasta Controle pode ser visto em: dicionário (figura 4.25); sintaxe (figura 4.22); frases (figura 4.26); classes (figura 5.11); e índices (figura 5.12).

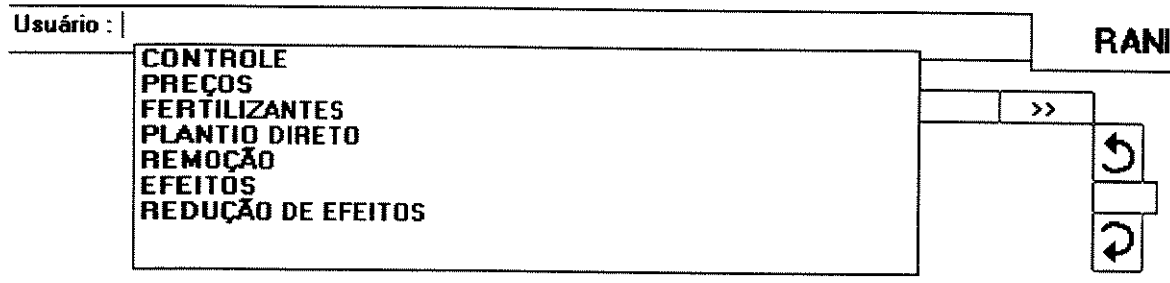

Figura 5.14: Interface do Kards mostrando as Pastas existentes na BD de Representações.

5.4.2 Comparação e Modificação

O processo de comparação e interação ocorre no sistema Kards. Um texto digitado ou inserido na BD tem seu conteúdo comparado com as representações existentes nas fichas Classe da BD de Representações. Quando é verificada a ocorrência de alguma das informações contida em alguma das fichas Classe, a parte do texto onde verificou-se a ocorrência é destacada (Figura 5.15) e a área à qual ela está associada é apresentada. Desta forma, o usuário é informado da ocorrência de um texto relacionado aos assuntos codificados na BD de Representações.

As comparações são direcionadas pela busca nos textos da ocorrência dos conceitos classificados como termos chave presentes nas interpretações contidas na ficha Classe. A verificação da ocorrência do termo chave dá início à verificação dos demais elementos da interpretação.

Os textos selecionados são então analisados, e dentro do contextos no qual o sistema foi implementado, são utilizados. Na análise, pode ocorrer, por exemplo, a verificação da necessidade da especialização ainda maior de uma área. Ou seja, a criação, à partir de uma única área de outras subdivisões. Neste caso, é necessário que para cada nova subdivisão seja definido o dicionário específico e a sintaxe que é utilizada na geração da estrutura. Uma outra situação consiste em verificar a necessidade da atualização da representação de alguma das áreas existente. Neste caso torna-se necessário a eliminação ou inserção de informações no dicionário e/ou nos símbolos da sintaxe da correspondente área. Em seguida é refeito a representação.

Conseqüentemente, as adaptações necessárias nas representações são feitas através da adaptação no dicionário e/ou na sintaxe.

As áreas que eventualmente perderem interesse são simplesmente eliminadas da BD de representações.

Nesta implementação, as representações utilizadas correspondem ao nível de conceitos complexos. A codificação das representações, dependendo da necessidade, pode requerer um nível de conhecimento correspondente ao nível de teoria ou até mesmo algum outro nível superior.

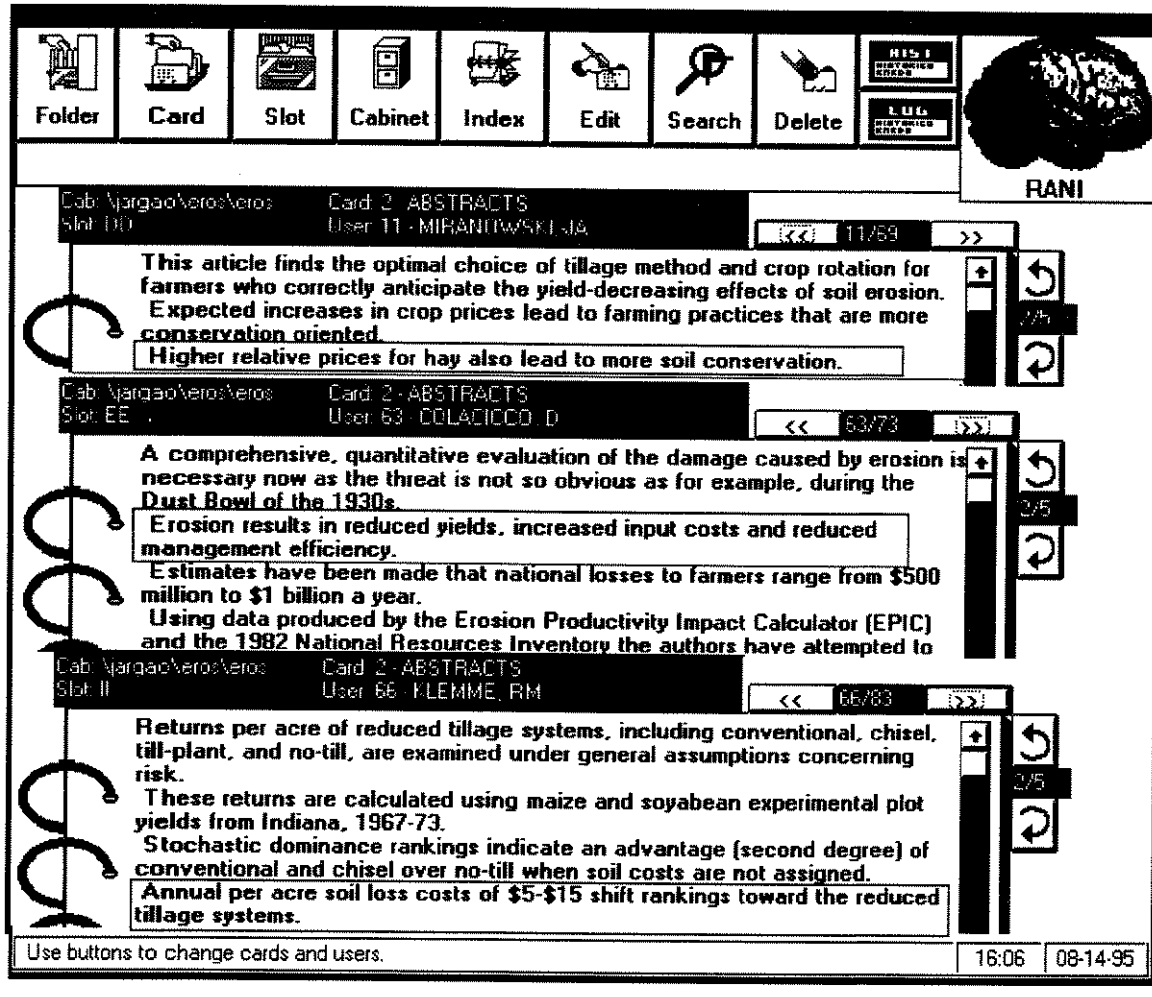

Figura 5.15: Filtragem dos Textos.

Um dos testes executados consistiu em obter um novo conjunto de resumos para verificar se as representações criadas com o conjunto inicial de resumos, permitiam a correta classificação. Assim, cada novo resumo foi apresentado e verificou-se a ocorrência de alguma representação (figura 5.15). Neste teste verificou-se uma correta associação entre as representações e os resumos classificados, entretanto não foi

verificado manualmente nos resumos não classificados a existência de alguma classificação incorreta.

A BD de representações pode também ser utilizada com a finalidade de ensino a iniciantes nas áreas nela relacionadas. Isto porque, o dicionário, a sintaxe e as representações permitem uma compreensão inicial das áreas de interesse, e os índices possibilitam o rápido acesso aos casos relacionados às áreas de interesse.

Outra utilização da estrutura apresentada consiste em implementar uma estrutura de Raciocínio Baseado em Casos. Os casos antigos em linguagem natural de alguma especialidade são utilizados na criação da BD de Representações. Nos novos casos apresentados ao sistema Baseados em Casos é verificada a existência de alguma das representações presentes na BD de representações. Quando alguma representação for encontrada, os índice correspondentes são utilizados para indexar os casos correlatos e desta forma, encontrar as soluções do referido caso.

CAPÍTULO 6

CONCLUSÕES

As ferramentas para manipular (extração de dados, recuperação de informações, etc) bases de dados com grande volume de textos e a filtragem das informações recebidas, certamente irão tornar-se de grande importância na vida dos usuários de computadores. Essas ferramentas são caracterizadas por necessitarem da especificação e da representação do conhecimento relacionado às áreas de interesses do usuário ou de um grupo de usuário. Portanto, este tipo de ferramenta deverá ser configurada por um usuário ou por um grupo, de acordo com a sua correspondente área de interesse. A codificação direta do conhecimento pelo usuário é a principal dificuldade no desenvolvimento de sistemas Baseados em Conhecimento, e que certamente ocorre na configuração dessas ferramentas de manipulação. Dentro deste contexto, o sistema Jargão é uma ferramenta que auxilia o usuário na construção da estrutura de conhecimento, através dos mecanismos de aprendizado existentes no sistema e do processo de adaptação feito pelo próprio usuário.

A utilização de analisadores sintáticos clássicos não é recomendado em sistemas para a análise de BDLN. A forma adotada tem sido a utilização dos chamados analisadores "fracos" ([JACO93]). Neste trabalho, o analisador sintático nebuloso apresentado possibilita ao usuário graduar a complexidade do analisador sintático de acordo com o interesse e o nível de conhecimento disponível. Conseqüentemente, o analisador nebuloso pode variar de um analisador sintático "fraco", através de uma sintaxe nebulosa simples, a um analisador sintático mais complexos através de uma sintaxe nebulosa complexa. No Jargão, um analisador clássico é especificado quando o usuário codifica todas as informações sintáticas a respeito do domínio em questão.

A complexidade da sintaxe está relacionada com a ambigüidade das interpretações existentes. Quanto mais simples é a sintaxe, maior é o número de interpretações existentes. Por outro lado, à medida em que a complexidade aumenta, menor é o número de interpretações.

Um aspecto importante na implementação de um sistema de análise refere-se ao léxico utilizado. A adoção de um dicionário com grande número de termos necessário para garantir a generalidade da análise, aumenta diretamente a complexidade

computacional baixando a relação de custo-benefício. No sistema Jargão, trabalha-se com a especialização, assim o léxico utilizado consiste dos termos presentes na BDLN, restringido de acordo com o contexto da especialização. Portanto, o léxico é consideravelmente menor e específico ao domínio em que está sendo utilizado. A especificação das classes sintáticas são também criadas pelo usuário e associadas de acordo com sua correspondência, aos termos do dicionário. O sistema foi concebido para permitir a reutilização de dicionário criado em outras análises. Entretanto, esta reutilização não ocorreu em virtude do sistema Jargão não estar sendo utilizado por um grande número de usuários.

Na arquitetura conexionista utilizada no Jargão, a rede neural RNE, o modelo do neurônio especificado tem como sua grande vantagem a incorporação dos aspectos de processamento numérico e simbólico, que permitem a sua utilização como um processador da linguagem. O processamento simbólico é feito através do processamento T^R>>C apresentado, feito através da especificação de uma linguagem nebulosa e dos correspondentes transmissores, receptores e controladores. O processamento numérico é codificado nas sinapses de acordo com a quantidade de processamento T^R>>C existente.

A topologia da RNE é gerada de acordo com a sintaxe nebulosa especificada e com o conhecimento contido na BDLN a ser analisada. Na representação de conhecimento lingüístico hierarquicamente organizado, as RNE podem ser hierarquicamente organizadas, e uma sintaxe é codificada para construção de cada um dos níveis de RNE. Neste trabalho, apresentamos uma estrutura hierárquica conceitual (sessão 2.4) formada por:

a) uma rede de nível inferior, a RNE de Conceitos Primitivos, especificada através de uma sintaxe primitiva, e que representa os conceitos primitivos;

b) uma rede de nível intermediário a RNE de Conceitos complexos, especificada através de uma sintaxe primitiva, e que representa os conceitos complexos;

c) e uma rede de mais alto nível, a RNE de Teorias.

Esta hierarquia conceitual mostrou-se suficiente para as aplicações desenvolvidas. Evidentemente, existem vários modelos lingüísticos hierárquicos que vêm sendo desenvolvidos, e que também poderiam ser especificados.

As informações codificadas no modelo conexionista utilizado podem ser lidas automaticamente de três formas diferentes: a quantitativa, a específica e a conceitual. A leitura quantitativa fornece a freqüência de uma informação na BDLN, e esta informação é obtida através da leitura do valor das sinapses. A leitura específica permite a obtenção das frases que satisfazem a sintaxe especificada, e que fornecem informações sobre a BDLN. A leitura conceitual fornece as relações sintáticas que descrevem as relações de conceitos suportadas pela BDLN. As informações obtidas através da leitura conceitual e

específicas são apresentadas aos usuários para a verificação da sua consistência com o contexto e para a definição da semântica associada.

Uma outra forma possível de leitura é a semântica que no Jargão é feita manualmente pelo usuário. A implementação da leitura semântica automática é possível quando adota-se analisadores semânticos, que requerem a especificação das primitivas semânticas (a maioria dos modelos semânticos utilizam as primitivas semânticas especificadas no formalismo da Dependência Conceitual) com as suas respectivas relações semânticas. A dificuldade de implementarmos a leitura semântica está no fato desta ser dependente das relações semânticas associadas a cada uma das primitivas semânticas, não podendo ser configurado de acordo com as semânticas especificadas pelo usuário. Nos problemas tratados utilizando o Jargão, as ambigüidades sintáticas e semânticas têm sido satisfatoriamente solucionadas manualmente pelo usuário.

O principal aspecto computacional a ser destacado no contexto do desenvolvimento das ferramentas de análise, consiste da necessidade das ferramentas estarem acopladas a um sistema de base de dados. O sistema Jargão está acoplado ao sistema de banco de dados Kards. Criou-se no Jargão mecanismos que permitem armazenar na base de dados do Kards as informações (dicionários, sintaxes, índices, etc) produzidas durante uma análise. Esta base de dados permite uma nova leitura das informações contidas na BDLN analisada.

As base de dados em linguagem natural geralmente são desorganizadas e dispõem de mecanismos ineficientes de indexação. O Jargão pode ser utilizado para obter da BDLN uma hierarquia conceitual que pode ser utilizada para criar uma estrutura de base de dados que permita a indexação conceitual dos textos. O Kards dispõe de uma estrutura de arquivos que permite a construção desta hierarquia conceitual. Como conseqüência as informações da BDLN podem ser indexadas de acordo com esta hierarquia conceitual. Na aplicação apresentada na sessão 5.4 (Filtragem de Informação), a criação da estrutura das representações consistem da implementação de uma hierarquia conceitual.

As ferramentas de análise, além de disporem de uma base de dados acoplada à ferramenta, devem dispor de mecanismos que permitam o acesso às informações de diversas estruturas de arquivos das outras BDLN.

Uma outra aplicação possível de ser feita utilizando o sistema Jargão consiste do resumo dos textos ou a padronização dos conceitos contidos na BDLN. Os resumos são gerados através da reescrita da BDLN com os conceitos específicos codificados pelo usuário. Desta forma, a base de dados reescrita torna-se especifica e padronizada de acordo com o contexto. A vantagem desta aplicação consiste em tornar a BDLN mais compacta, sem os textos que não estão relacionado com o contexto. Na recodificação da

BDLN é fornecido também a freqüência com que um conceito é reescrito, permitindo desta forma a medida quantitativa das informações da BDLN.

O sistema Jargão foi utilizado, em diferentes BDLN, como uma ferramenta de aquisição de conhecimento, permitindo a especificação de uma metodologia de aquisição de conhecimento apresentada no capítulo 5. Uma das vantagens apresentadas neste sistema trata-se do fato da estrutura de representação de conhecimento na qual o conhecimento é codificado não ser fixa. A especificação da estrutura é feita através da sintaxe.

A principal dificuldade na aquisição do conhecimento consistiu, alguma vezes, da pobreza de detalhes e da ausência das informações nos textos. A verificação das características do conteúdo da BDLN permite, por exemplo, que seja verificado a real condição para a construção do sistema baseados em casos.

A extração de padrões utilizando o sistema Jargão foi um aspecto pouco explorado, isto ocorreu fundamentalmente por não encontrarmos problemas práticos que pudéssemos aprimorar esse aspecto do sistema. Na implementação feita, mostrada na sessões 4.8, utilizou-se os conceitos de operadores e a cada um deles foi associado alguma ação. Os testes consistiram da extração dos padrões numéricos presentes em uma BDLN. Neste contexto, a sugestão para aprimorar o sistema consiste da especificação através da sintaxe dos operadores e das ações correspondentes. Desta forma, estaria implementado no neurônio o processamento da tradução da mensagem (Figura 2.7), que consiste em produzir os controladores que estariam associados às ações especificadas. Evidentemente, a implementação estaria incorporada ao algoritmo de geração da RNE.

Os sistemas apresentados são claramente associados ao casamento de padrões. A principal sugestão consiste em criar sintaxe correspondentes as noções relacionadas à Dependência Conceitual, que permitiria o desenvolvimento de processadores semânticos.

BIBLIOGRAFIA

[BELK92] Nicholas J. Belkin and W. Bruce Croft, Information Filtering and Information Retrievel: Two Side of Same Coin?, *Communications of ACM*, Vol. 35, Nº 12, pag. 29-38, December, 1992.

[BUCH83 et alli] B. Buchanan, D. Barstow, R. Bechtal, J. Bennett, W. Glancey, C. Kulikowski, T. Michel, and D. Waterman, Constructing an Expert System, In F. Hayes-Roth, D. Watterman, and D. Lenat, Eds. *Building Expert Systems*, Readings, MA: Addison-Wesley, 1983, 748 pag.

[CHOM57] Noam Chomsky, *Syntactic Structures*, the Hague: Mouton, USA, 1957.

[CHOM65] Noam Chomsky, *Aspects of the theory of syntax*, MIT Press, Cambridge, USA, 1965, 251 pag.

[DAMA92] António R. Damásio and Hanna Damásio, Brain and Language, *Scientific American*, September, 1992.

[FIRE88] Morris W. Firebaugh, *Artificial Intelligence: An Knowledge Based Approach,* Boyd & Fraser Publishing Company, Boston, 1988, 740 pag.

[GUIL94] Ivan R. Guilherme, A. F. Rocha, Aquisição de Conhecimento de Textos utilizando técnica conexionista, *Anais do Primeiro Congresso Brasileiro de Redes Neurais*, Itajuba, Outubro, 1994.

[GUIL95] Ivan R. Guilherme and M. T. Rocha, Neural Tools for Conceptual Data Analysis, *Proceedings of 6th International Fuzzy Systems Association World Congress*, Vol 1, pag. 693-696, São Paulo, 1995.

[HOPC69] J. E. Hopcroft and J. D. Ullman, *Formal Languages and Their Relation to Automata*, Addison Wesley, Reading Mass., 1969, 242 pag.

[JACO90] Paul S. Jacobs and Lisa F. Rau, SCISOR: Extracting Information from on-line news, *Communications of ACM*, Vol. 33, Nº 11, pag. 88-97, 1990.

[JACO93] Paul S. Jacobs and Lisa F. Rau, Inovations in text interpretation, *Artificial Intelligence*, Vol 63, Nº 1-2, pag. 143-191, 1993.

[JOHN92] Mark F. St. John, The Story Gestalt: A Model of Knowledge-Intensive Processes in Text Comprehension, *Cognitive Science*, Vol. 16, Nº 2, pag. 271-306, 1992.

[KAUF75] A. Kaufman, *Introduction to the theory of fuzzy Subsets*, Academic Press, New York, 1975, 600p.

[LENA90 ett all] Douglas B. Lenat, Ramanathan V. Guha, Karen Pittman, Dexter Pratt, and Mary Shepherd, CYC: Toward Programs with Common Sense, *Communications of ACM*, Vol. 33, Nº 8, pag. 30-49, August, 1990.

[LI95] Hang Li and Naoki Abe, Generalizing Case Frame Using a Thesaurus and MDL Principle, C&C Res. Lab, NEC, August, 1995.

[LYON87] John Lyons, *Linguagem e Linguística - Uma Introdução*, Editora Guanabara, Ed. Guanabara, Rio de Janeiro, 1987, 322p.

[MAUL91] Michael L. Mauldin, Retrievel Performance in FERRET: A Conceptual Information Retrievel System, *The 14Th International Conference on Research and Development in Information Retrievel*, Chicago, October, 1991.

[MACH91] Ricardo J. Machado, Armando F. Rocha, and Ivan R. Guilherme, FRANK: A Hybrid Fuzzy Connectionist and Baysian Expert System, *Proceedings of 4th International Fuzzy Systems Association World Congress*, Vol. 3, pag. 125-128, Bruxelas, Belgium, 1991.

[McCL86] J. L. McClelland and D. E. Rumelhart, *Parallel Distributed Processing: Explorations in Microstructure of Cognition* (Vol. 2), Cambridge, MA: Bradford Books, 1986, 611 pag.

[McGR89] K. L. McGraw and K. H. Bridggs, *Knowledge Acquisition: Principles and Guidelines*, Englewood Cliffs, Prentice Hall, 1989, 371 pag.

[MIIK91] Risto Miikkulainen and Michel G. Dyer, Natural Language Processing with Modular PDP Networks and Distributed Lexicon, *Cognitive Science*, Vol. 15, Nº 3, pag. 343-391, 1991.

[MIUR91] Kazuo Miura, Armando F. Rocha, and Ivan R. Guilherme, Knowledge acquisition from natural language data bases, in *Proc. LAIC-PEP'91- Latin American Conference on Artificial Intelligence in Petroleum Exploration and Production*, Rio de Janeiro, Brasil, 1991.

[MIZU73 Ett all] Masaharu Mizumoto, Junichi Toyoda, and Kohkichi Tanaka, N-Fold Fuzzy Grammars, *Information Sciences*, 5, pag. 25-43, 1973.

[ORNS91] R. E. Ornstein, *The Evolution of Consciousness*, Touchstone Simon & Schuster, New York, 1991.

[PARS89] K. Parsaye, Machine Learning: the Next Frontier, *PC - AI*, July/August, Vol 3, Nº 4, pag. 26-32, 1989.

[PATR92] Antonio R. Patricio, "Um sistema especialista para apoio a operação de plantas marítmas de processo", Dissertação de Mestrado apresentada na FEM-Unicamp, Novembro, 1992.

[PERE80] F. C. N. Pereira, and D. H. Warren, Definitive clause grammars for language analysis - a survey of the formalism and a comparison with argumented transition networks, *Communications of ACM*, Vol. 13, Nº 3, pag. 231-278, 1980.

[RAM92] Ashwin Ram, Natural Language Understanding for Information-Filtering Systems, *Communication of ACM*, Vol. 35, Nº 12, pag. 80-81, December, 1992.

[ROCH92a et all] A. F. Rocha, I. R. Guilherme, M. Theoto, A. M. K. Miyadahira, and M. S. Koizumi, A Neural Network for extracting Knowledge from Natural Language Data Bases, *IEEE Transactions on Neural Network*, Vol. 3, Nº 5, September 1992.

[ROCH92b et all] Armando F. Rocha, Ivan R. Guilherme, and Ricardo J. Machado, Knowledge Acquisition: A Connectionist Approach, *Proceedings of 3th Annual Simposiun of the International Association of Knowledge Engeenering- IAKE*, November, 1992, Washington, USA.

[ROCH92c] Armando F. Rocha, *The theory of Brains and Machines*, in Lectures and Notes in Artificial Intelligence, New York: Springer Verlag, 1992, 400pp.

[ROCH95] Armando F. Rocha, Ivan R. Guilherme, and Marcelo T. Rocha, Neural Systems and Symbolic Processing, Submetido, 1995.

[ROCH94] Marcelo T. Rocha, Armando F. Rocha, and Ivan R. Guilherme, Using Neural Tools in Environmental Data Analysis, *Proceedings of 3rd International Conference on Fuzzy Logic, Neural Network and Soft Computing*, Iizuka, Japan, August, 1994.

[ROCH94] Marcelo T. Rocha, Armando F. Rocha, and Ivan R. Guilherme, Neural Systems and Concepts Reasoning, *Proceedings of Second European Congress on Intelligent Techniques and Soft Computing*, pag. 376-377, Aachen, Germany, September, 1994.

[ROSE91] Daniel E. Rose and Richard K. Belew, A Connectionist and Symbolic Hybrid for Improving Legal Research , *International Journal Man-Machine Studies*, Vol. 35, pag. 1-33, 1991.

[RUME86] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed processing: Explorations in Microstructure of Cognition* (Vol. 1), Cambridge, MA: Bradford Books, 1986, 547 pag.

[SAGE87 Ett all] Naomi Sager, Carol Friedman, and Margaret S. Lyman, *Medical Language Processing - Computer Management of Narrative Date*, Addison-Wesley, New York, 1987.

[SATO92] Ademar T. Sato, "Sistema Inteligente para elaborar um projeto de perfuração de um poço de petróleo", Dissertação de Mestrado apresentada na FEM-Unicamp, Dezembro, 1992.

[SHAN75] R. C. Shank and R. P. Abelson, *Scripts, plans, goals and understanding: An inquiry into human knowledge structure*, Hillsdale, NJ:Erlbaum, USA, 1977.

[SHAN82] Roger C. Shank, *Dynamic Memory*, New York: Cambridge University Press, USA, 1982.

[SHAS88] Lokendra Shastri, A Connectionist Approach to Knowledge Representation and Limited Inference, *Cognitive Science*, Vol. 12, Nº 3, pag. 331-392, 1988.

[SODE94] Stephen Soderland and Wendy Lehnert, Corpus-Driven Knowledge Acquisition for Discourse Analysis, *Proceeding of the Twelfth National Conference on Artificial Intelligence*, 1994.

[SOWA84] John F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, USA, 1984.

[SLAT92] Brian M. Slator, Sense and Preference, *Computers Mathematics with Applications*, Vol 23, Nº 6-9, pag. 391-402, 1992.

[WALT85] D. L. Waltz and J. B. Pollack, Massively parallel parsing: a strongly interactive model of natural language interpretation, *Cognitive Science*, Vol. 9, Nº 1, pag. 51-74, 1985.

[WINO72] Terry Winograd, *Understanding Natural Language*, Academic Press, New York, 1972, 191 pag.

[WOOD70] W. A. Woods, Transition networks grammars for natural language analysis, *Communications of ACM*, Vol. 13, Nº 10, pag. 591-606, August, 1970.

[WOOD73] W. A. Woods, Progress in Natural Language Understanding: An Applications to Lunar Geology, *AFIPS Conference Procedings*, 42, pag. 441-450, AFIPS Press Montvale, 1973.

[ZADE65] Lotfi A. Zadeh, Fuzzy Sets, *Information and Control*, Vol. 8, pag. 338-353, 1965.

[ZIME91] H. J. Zimmerman, *Fuzzy Set Theory - and its Applications*, Kluwer Academic Publisher, USA, 1991.