

**Análise automática de
sumários em língua
portuguesa: uma aproximação
ao tratamento da estrutura de
um texto**

Horacio Saggion

Análise automática de sumários em língua portuguesa: uma aproximação ao tratamento da estrutura de um texto

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida pelo Sr. Horacio Saggion e aprovada pela Comissão Julgadora.

Campinas, 30 de outubro de 1995.

Ariadne M. B. R. Carvalho
Profa. Dra. Ariadne Maria Brito Rizzoni
Carvalho
Orientador

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

UNIDADE	BC
N.º CHAMADA:	
V.	F.
TIPO DE	26.211
PREÇO	433,95
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	02/12/95
N.º CPU	

CM-00081050-7

FICHA CATALOGRAFICA ELABORADA PELA
BIBLIOTECA DO INECC DA UNICAMP

Saggion, Horacio

Sa18a Analise automatica de sumarios em lingua portuguesa : uma
aproximacao ao tratamento da estrutura de um texto/ Horacio
Saggion. -- Campinas, [SP : s.n.], 1995.

Orientador : Ariadne M.B.R. Carvalho.

Dissertacao (mestrado) - Universidade Estadual de Campinas,
Instituto de Matematica, Estatistica e Ciencia da Computacao.

1. Anafora (Linguistica). 2. Linguistica - Processamento de
de dados. 3. Analise (Gramatica por Computador). I. Carvalho,
Ariadne M.B.R. II. Universidade Estadual de Campinas, Instituto
de Matematica, Estatistica e Ciencia da Computacao. III. Titulo.
Ciencia da Computacao. III. Titulo.

Tese de Mestrado defendida e aprovada em 30 de OUTUBRO de 1995
pela Banca Examinadora composta pelos Profs. Drs.

Vera Lúcia Strube de Lima

Prof (a). Dr (a). VERA LÚCIA STRUBE DE LIMA

Jacques Wainer

Prof (a). Dr (a). JACQUES WAINER

Ariadne M. B. R. Carvalho

Prof (a). Dr (a). ARIADNE MARIA BRITO RIZZONI CARVALHO

Análise automática de sumários em língua portuguesa: uma aproximação ao tratamento da estrutura de um texto¹

Horacio Saggion²

Departamento de Ciência da Computação
IMECC – UNICAMP

Banca Examinadora:

- Ariadne M.B.R. Carvalho (Orientadora)³
- Vera Lúcia Strube de Lima⁴
- Jacques Wainer³
- Jorge Stolfi (Suplente)³

¹Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação da UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

²O autor é Licenciado em Ciência da Computação pela Universidade de Buenos Aires.

³Professor do Departamento de Ciência da Computação - IMECC - UNICAMP.

⁴Professora do Instituto de Informática - PUC - RS.

À memória do meu pai Mário, um exemplo de progresso.

A minha mãe Sima.

A minha esposa Sandra.

Agradecimentos

A minha orientadora Ariadne pela confiança no meu trabalho e sobre tudo pela sua amizade nestes anos.

A minha esposa Sandra, que me apoio incondicionalmente nesta aventura brasileira.

A minha doce irmã Clarisa.

A minha mãe Sima.

A minha família.

A meus amigos Oscar e Adriana pela companhia constante.

A minha família da rua Villarroel.

Aos amigos Rogelio e Lia, sempre conosco.

Aos amigos Mariano, Maria e Ana Maria que com sua amizade fazem a vida mais fácil.

A Luiz Cláudio Rosa sempre de coração contente.

Aos meus colegas das turmas 1993, 1994 e 1995.

A Clevan, Mateus, “la Petty”, Diego, Fabio, Helvio, Daniela, Anderson, Humberto, Pedro Rafael, Victor, Cris, Cristina, Ricardo, Elaine, Sueli, Márcio Botelho, Márcio Stefani, Alexandre, Raul, Mauricio, Rosiane, Eduardo, Nuccio, Karem, Marques, George, Arturo, Mario, Helena, Juliano, Luis Mariano, Cereja.

Aos meus professores do Departamento de Ciência da Computação Ariadne Carvalho, Célia Picinin de Mello, Heloisa Vieira da Rocha, Jacques Wainer, Marcus Vinicius Poggi de Aragão, Nelson Machado, Pedro Rezende, Ricardo Dahab, Tomasz Kowaltowski e da Linguística Carlos Franchi.

A Ricardo Dahab, em uma pessoa só o melhor do brasileiro.

Ao professor Jorge Stolfi pelos comentários ao meu trabalho.

Aos funcionários do departamento de computação, em especial a Alda, Solange e Roseli.

Aos funcionários da biblioteca do IMECC, em especial ao Carlos e Luci.

Aos funcionários da biblioteca do IEL, em especial a Belkis.

Aos funcionários do IMECC.

A Lúcia Rino pelas suas valiosas sugestões e discussões e pelo extenso material bibliográfico que tanto contribuiu para este trabalho.

Ao professor Cláudio Luchessi, coordenador de pós-graduação neste período.

Aos amigos da UNICAMP.

À Universidade Estadual de Campinas que tornou possível o desenvolvimento deste trabalho.

Ao Departamento de Computação da Universidade de Buenos Aires e muito especialmente a Irene Loiseau que me apoio nesta etapa.

Ao CNPq e à FAPESP pelo apoio financeiro recebido.

Ao Brasil, terra maravilhosa, que nos recebe de braços abertos.

Resumo

Lingüística Computacional é a ciência que estuda as teorias que possibilitam a construção de sistemas capazes de entender e gerar linguagem natural. Muitos sistemas de entendimento e, em particular, muitos sistemas de tradução automática concentram seus esforços no tratamento da sentença como unidade lingüística fundamental. Um texto é concebido como uma justaposição de sentenças e, de acordo com essa concepção, a tradução é feita sentença por sentença, muitas vezes desconsiderando fenômenos que caracterizam um texto como tal. Nesta dissertação estamos concentrados na análise de um texto em língua portuguesa via computador, visando o tratamento de fenômenos de interesse para a tradução automática. Devido a complexidade do tratamento de textos irrestritos, decidimos trabalhar com sumários de artigos técnicos pois, embora curtos, são considerados textos e, como tal, apresentam os fenômenos de interesse para o nosso trabalho. A nossa principal preocupação é o entendimento da estrutura de um sumário e o processo de geração dessa estrutura. Acreditamos que a representação da estrutura de um sumário possa auxiliar na produção de uma tradução de alta qualidade.

No processamento de um texto e no cálculo de sua estrutura dois estudos devem ser considerados: a coesão e a coerência textual. Esses estudos tentam responder ao problema da conexidade das sentenças na seqüência lingüística. Nesta dissertação abordamos o problema da coesão textual como mecanismo de vinculação de sentenças no texto. O principal fenômeno abordado é a coesão referencial através do estudo e tratamento de anáfora definida.

Apresentamos um formalismo para representar um subconjunto de sumários em língua portuguesa e um processo computacional que, a partir da análise do sumário gera a representação proposta nesta dissertação. O formalismo reflete o conteúdo proposicional do sumário e as relações entre proposições. O sistema analisa as sentenças e as vincula na representação proposta. Acreditamos que tanto a representação proposta como o processo de análise textual possam ser estendidos para outros domínios textuais.

Abstract

Computational Linguistics studies theories that enable the construction of systems capable of understanding and generating natural language. Many understanding systems and, in particular, many automatic translation systems, treat sentences as the fundamental linguistic unit. A text is conceived as a juxtaposition of sentences and, according to this conception, translation is made sentence by sentence; many times phenomena that characterize the text are ignored.

The work presented here is concerned with the automatic analysis of texts in the Portuguese language dealing with phenomena of interest for automatic translation. Due to the complexity of the treatment of unrestricted texts, we have worked with abstracts of technical papers; although they are short, they are still texts and, as such, present phenomena of interest for our work.

Our main concern is the understanding of the abstract's structure and the generation of such a structure. We believe that this representation may help in the production of high quality translation. When processing a text and calculating its structure two phenomena must be taken into account: cohesion and coherence. These studies try to explain the connection between sentences. Our work takes cohesion into account in order to make the connection between sentences in the text. The main phenomenon treated here is definite anaphora.

We present a formalism to represent a subset of abstracts in Portuguese and a computational process through which this representation is produced. The formalism reflects the propositional content of the abstract and the relationships between propositions. The sentences are analysed and related to each other in the proposed representation. We believe that the representation and the text analysis proposed may be extended to other textual domains.

Conteúdo

1	Introdução	1
1.1	O problema	2
1.2	Organização da dissertação	3
2	O domínio da tradução automática	4
2.1	Componentes de um sistema de TA	4
2.2	Arquiteturas de sistemas de TA	5
2.3	O domínio dos sumários	8
2.3.1	Tipos de sumários	9
2.3.2	Estrutura de um sumário	10
2.3.3	Usos e aplicações dos sumários	11
2.4	O corpus	12
2.5	Conclusões	13
3	Processamento Textual	14
3.1	A estrutura textual	14
3.2	Abordagens à estrutura textual	15
3.3	A coerência textual	16
3.4	A coesão textual	19
3.4.1	A coesão referencial	19
3.4.2	Exófora e endófora	20
3.4.3	Anáfora e catáfora	21
3.4.4	Anáfora profunda	22
3.5	Formas abreviadas no português	23
3.5.1	Elipse	23
3.5.2	Pronomes	23
3.5.3	Frase nominal definida	24
3.5.4	Outras formas abreviadas na língua portuguesa	25
3.6	Restrições sintáticas à referência pronominal	26
3.7	Tratamento computacional de referência e co-referência	29
3.7.1	Teorias sintáticas na resolução de pronomes	32
3.7.2	Necessidade de resolução de frases nominais definidas	33
3.8	A coesão referencial no corpus	36
3.9	Conclusões	40

4	Representando a estrutura de um sumário	42
4.1	Componentes da representação	45
4.2	Representação das entidades	48
4.2.1	Representação das anáforas	53
4.3	Representação das proposições	53
4.4	Representação de nominais derivados de um verbo	55
4.5	Representação dos advérbios	55
4.6	Representação das relações proposicionais	56
4.7	Representação dos segmentos	59
4.8	Exemplos de proposições e relações	59
4.9	Representações	62
4.10	Conclusões	66
5	Construção da representação de um sumário	67
5.1	Construindo entidades e proposições	68
5.1.1	Referência e co-referência	69
5.1.2	Analisando expressões nominais	71
5.1.3	Proposições e relações entre proposições	73
5.2	Identificação dos segmentos	78
5.3	Arquitetura do Analisador Textual	81
5.3.1	Iniciando a análise	86
5.3.2	Análise das sentenças	88
5.3.3	Resolvendo pronomes	95
5.3.4	Interpretando frases nominais	97
5.3.5	Construindo a estrutura textual	100
5.4	Conclusões	105
6	Conclusões	106
A	Exemplos de análise textual	110
A.1	Exemplo I	110
A.2	Exemplo II	112
A.3	Exemplo III	113
A.4	Exemplo IV	114
B	Código do analisador	116
	Bibliografia	161

Lista de Tabelas

3.1	Padrões de referência e co-referência no corpus	41
4.1	Padrões nominais encontrados no corpus de sumários	48
4.2	Padrões verbais encontrados no corpus de sumários	49
4.3	Padrões sentenciais simples encontrados no corpus de sumários . .	49
4.4	Padrões sentenciais compostos encontrados no corpus de sumários	49
4.5	Mapeamento dos elementos lexicais	61

Lista de Figuras

2.1	Sistema de tradução direto	6
2.2	Sistema de tradução baseado em entendimento	6
2.3	Sistema transfer	7
2.4	Sistema interlíngua	7
2.5	Sistema tradutor de sumários	9
2.6	Os sumários: documentação e lingüística	9
2.7	Sumários indicativos	10
2.8	Sumário informativo	10
2.9	Os sumários e os leitores	11
2.10	Um sumário do corpus (1)	12
2.11	Um sumário do corpus (2)	13
3.1	Coerência em texto (1)	16
3.2	Representando sentenças (1)	17
3.3	Banco de conhecimentos (1)	17
3.4	Relação de contraste em texto	17
3.5	Representando sentenças (2)	18
3.6	Conhecimento necessário para a verificação da relação de contraste	18
3.7	Relação de Elaboração em texto	18
3.8	Coerência em texto (2)	19
3.9	Scripts	19
3.10	A coesão referencial	20
3.11	Exófora	20
3.12	Anáfora	21
3.13	Catáfora	21
3.14	Referência e co-referência (1)	21
3.15	Anáfora profunda	22
3.16	Referência e co-referência (2)	22
3.17	Elipse	23
3.18	Pronomes demonstrativos	24
3.19	Anáfora definida em texto (1)	25
3.20	Anáfora definida em texto (2)	25
3.21	Outras formas de referência abreviada	26
3.22	Pronomes e categorias sintáticas	27
3.23	Restrições no nível sentencial	29

3.24	C-comando (1)	29
3.25	C-comando (2)	30
3.26	Pronomes e tradução automática	30
3.27	Segmentos e sub-segmentos	32
3.28	Resolução dos pronomes	33
3.29	Frases definidas e tradução automática	34
3.30	Frases definidas e entendimento (1)	34
3.31	Frases definidas e entendimento (2)	34
3.32	Um mundo de blocos	35
3.33	Representação do mundo dos blocos	35
3.34	Referência anafórica em sumários (1)	36
3.35	Referência anafórica em sumários (2)	37
3.36	Referência anafórica em sumários (3)	37
3.37	Referência anafórica em sumários (4)	37
3.38	Referência não anafórica em sumários (1)	38
3.39	Referência não anafórica em sumários (2)	38
3.40	Entidades do contexto situacional	39
3.41	Referência pronominal em sumários (1)	39
3.42	Referência pronominal em sumários (2)	39
3.43	Referência pronominal em sumários (3)	40
4.1	Tipo de informação em sumários	43
4.2	Representação esquemática da estrutura textual	43
4.3	Analisando um sumário	45
4.4	Representação parcial de um sumário	45
4.5	Segmentos no texto	60
4.6	Segmento de sumário	61
4.7	Representação de um segmento de sumário	62
4.8	Exemplo de sumário (1)	63
4.9	Representação do sumário (1)	63
4.10	Exemplo de sumário (2)	64
4.11	Representação do sumário (2)	65
4.12	Exemplo de sumário (3)	65
4.13	Representação do sumário (3)	65
5.1	Arquitetura do analisador textual	67
5.2	Componentes da representação	68
5.3	Obtenção de um referente	69
5.4	Uma representação	70
5.5	Criação de um referente	70
5.6	Frases nominais em um texto	71
5.7	Sintagma nominal (1)	72
5.8	Sintagma nominal (2)	72
5.9	Sintagma nominal (3)	73

5.10	Sintagma nominal (4)	74
5.11	Análise sintática (1)	75
5.12	Análise sintática (2)	75
5.13	Análise sintática (3)	76
5.14	Análise sintática (4)	77
5.15	Exemplo de Sumário (1)	79
5.16	Exemplo de Sumário (2)	80
5.17	Exemplo de Sumário (3)	81
5.18	Marcadores de segmentos no corpus	82
5.19	Arquitetura do analisador sentencial	83
5.20	Arquitetura do montador da estrutura textual	85
5.21	Arquivo contendo um sumário a ser analisado	87
5.22	Codificação dos tokens	87
5.23	Uma gramática	89
5.24	Uma gramática em DCG	89
5.25	Uma gramática em DCG com parâmetros	90
5.26	Uma Slot Grammar	91
5.27	Análise de um nominal com valor predicativo	92
5.28	Vinculação de sentenças (1)	104
5.29	Vinculação de sentenças (2)	104
6.1	O trabalho desenvolvido	107
6.2	Extensões	109

Capítulo 1

Introdução

A linguagem é um sistema maravilhoso de expressão que faz com que seja possível a codificação de idéias, sentimentos e desejos em uma cadeia sonora ou escrita, fornecendo o meio ideal de comunicação humana e o principal sistema para a preservação do conhecimento.

Embora o cérebro humano seja a única máquina capacitada para entender e gerar linguagem, a idéia de utilizar um computador como instrumento de entendimento não é nova. A Linguística Computacional é a Ciência que estuda as teorias que possibilitam a construção de sistemas computacionais capazes de entender e gerar linguagem natural. Sendo esse o principal meio de comunicação entre os homens, existe a idéia de utilizá-lo também na comunicação entre o homem e o computador, para possibilitar o acesso “natural” à informação armazenada nos sistemas de informação. A necessidade de disseminação de informação expressa em linguagem natural faz com que seja necessário o desenvolvimento de sistemas capazes de traduzir documentos para outras línguas. A agregação de países em comunidades políticas e econômicas demanda hoje o desenvolvimento de sistemas capazes de gerar documentos em várias línguas.

Além do interesse nestas áreas aplicativas, o desenvolvimento de sistemas em linguística computacional possibilita o teste de teorias linguísticas, psicológicas e cognitivas, sobre o complexo processo de entendimento e produção de linguagem natural.

A língua apresenta organização em todos os seus níveis, desde as palavras e as sentenças, até o agrupamento destas últimas em textos coerentes. Embora as primeiras aplicações da linguística computacional tenham se concentrado no domínio da sentença, há tempos procura-se ultrapassar essa barreira visando atacar o problema da expressão linguística a nível textual. Para processar adequadamente um texto é necessário entender os princípios que governam essa estrutura linguística e que diferem da estrutura linguística da frase.

A estrutura de um texto se refere aos princípios que governam a agrupação de sentenças em unidades coerentes. O cálculo dessa estrutura é fundamental em qualquer sistema de processamento de linguagem natural. Notamos, entretanto, que os sistemas de tradução automática tratam o texto como uma justaposição de sentenças, muitas vezes desconsiderando fenômenos que ultrapassem a barreira sentencial. Acreditamos que o conhecimento sobre a estrutura textual é essencial para a obtenção de uma tradução de alta qualidade.

1.1 O problema

Nesta dissertação estamos concentrados no problema da análise de um texto em língua portuguesa via computador, sendo o nosso principal objetivo o tratamento de fenômenos de interesse para a tradução automática. Para isso, restringimos o domínio textual a sumários de artigos técnicos¹ pois, apesar de muitas vezes serem restritos a um único parágrafo, são considerados textos e, como tal, possuem os fenômenos textuais de interesse para o nosso trabalho.

Nesta dissertação abordamos o problema do entendimento da estrutura de sumários de artigos técnicos. Para isso estudamos a sua estrutura e propusemos um formalismo para representar um subconjunto de sumários em língua portuguesa. Especificamos um sistema de processamento de linguagem natural que faz o mapeamento de um sumário para uma representação textual. Esse sistema poderá fazer parte de um sistema de tradução automática de sumários do português para outras línguas.

A representação proposta foi obtida a partir do estudo de textos reais e da verificação de relações entre sentenças. A representação consta de três partes fundamentais: (i) um conjunto de segmentos textuais; (ii) um conjunto de proposições e entidades; e (iii) relações que vinculam os segmentos do texto. Para o processamento de um texto e cálculo da sua estrutura dois estudos direcionados pela lingüística textual são geralmente considerados: (i) a coesão textual; e (ii) a coerência textual. Esses estudos procuram tratar o problema da conexidade das sentenças no texto. Nesta dissertação estamos propondo o tratamento de apenas um fenômeno de coesão textual, conhecido como anáfora definida. A resolução de anáforas definidas é fundamental em qualquer sistema de entendimento de linguagem natural; em particular, é fundamental na construção da representação textual proposta nesta dissertação.

¹Nesta dissertação a palavra *sumário* será considerada um sinônimo da palavra *resumo* e da palavra inglesa *abstract*.

1.2 Organização da dissertação

No Capítulo 2 introduzimos os conceitos relativos à tradução de línguas por computador e também apresentamos o domínio textual escolhido. No Capítulo 3 introduzimos os conceitos sobre coesão e coerência textuais. No Capítulo 4 definimos uma representação para um subconjunto de sumários em língua portuguesa. No Capítulo 5 apresentamos o sistema de processamento de sumários proposto e os aspectos de implementação de um protótipo experimental. Finalmente no Capítulo 6 apresentamos as conclusões e contribuições advindas deste estudo.

Capítulo 2

O domínio da tradução automática

A palavra *tradução* é utilizada para descrever toda tarefa pela qual uma expressão lingüística em uma língua, chamada *língua origem* (*LO*), é transformada em uma expressão lingüística em outra língua, chamada *língua destino* (*LD*), de maneira tal que as expressões sejam “equivalentes”. A entidade humana ou artificial responsável pela tarefa de traduzir é chamada de *tradutor*. Um tradutor deve conhecer tanto a *LO* quanto a *LD* e deve entender também o campo de conhecimento da expressão a ser traduzida. O principal problema enfrentado por um tradutor é a inexistência de equivalências exatas entre expressões das línguas envolvidas no processo [Crystal 1987]. Quando a entidade encarregada da tarefa de traduzir é um computador utilizamos a expressão *tradução mecânica* ou *tradução automática* (TA).

2.1 Componentes de um sistema de TA

Os sistemas de TA podem ser divididos em dois grandes grupos: (i) sistemas diretos, que incorporam em um único programa os conhecimentos necessários para o processo de tradução completa; e (ii) sistemas indiretos, que dividem o processamento em estágios independentes (*análise* do fenômeno lingüístico na *LO* e *síntese* ou *geração* na *LD*).

Os componentes básicos de um sistema de TA são:

- *Dicionários:*

Contém informação necessária para os passos de análise e síntese. Dependendo da arquitetura escolhida, um dicionário ou vários serão utilizados. Sistemas diretos em geral utilizam-se de um dicionário bilíngüe, onde as palavras da língua origem estão relacionadas com o seu equivalente na língua destino. Sistemas indiretos têm em geral vários dicionários: unilíngüe para a análise de cada texto na língua origem e para a síntese na língua destino e bilíngüe para o estágio de conversão entre a análise e a síntese. Os

dicionários podem conter todas as palavras da língua ou podem conter só informação morfológica de palavras regulares da língua. As formas irregulares apareceram listadas de forma exaustiva.

- *Analisador Léxico-Morfológico:*

Cada sentença do texto original deverá ser dividida em uma seqüência de palavras. O processo de identificação das palavras pode ser muito simples, como no caso do português, língua na qual a divisão é dada pelos espaços em branco entre cada uma das unidades, ou um pouco mais complicado, como em línguas que utilizam-se de marcadores especiais para delimitar as palavras, ou que não utilizam marcas explícitas. Cada palavra da sentença deverá ser processada para que se obtenha a sua categoria gramatical e seus traços sintáticos e semânticos. O processo pode ser uma simples busca em dicionário, ou pode requerer uma análise morfológica.

- *Analisador Sintático-Semântico:*

Uma vez que cada palavra tenha a sua categoria gramatical e seus traços sintáticos definidos, a estrutura da frase pode ser determinada. A análise é geralmente baseada numa gramática da língua que descreve as frases aceitáveis. O processo de análise produz uma *árvore de análise* que será posteriormente utilizada pelo analisador semântico. Este último gera uma representação do significado da sentença a partir da informação semântica obtida dos itens lexicais.

- *Conversor:*

Em sistemas indiretos o conversor é o encarregado de trocar os itens lexicais da *LO* pelos itens lexicais da *LD*. Se a representação é de tipo sintática (geralmente uma árvore de análise), então o conversor, além de fazer as mudanças lexicais, re-arranjará a árvore de modo que a estrutura coincida com a estrutura da língua destino.

- *Gerador:*

Finalmente, a tradução na língua destino é produzida. Em sistemas com representações de tipo sintática, o processo consiste na linearização da árvore e na síntese das palavras. Em sistemas com representações de tipo semântica deve-se levar em consideração a escolha da estrutura da frase a gerar, a escolha de itens lexicais, e finalmente a síntese.

2.2 Arquiteturas de sistemas de TA

Os primeiros sistemas (diretos) foram projetados para a tradução entre um par de línguas definidas no início do projeto, correspondendo à arquitetura apresentada na Figura 2.1. Um único programa incorporava as fases de análise e geração.

Segundo [Hutchins 1984] a característica marcante desses sistemas era a ausência de conhecimentos sobre os fenômenos lingüísticos; a idéia era que utilizando

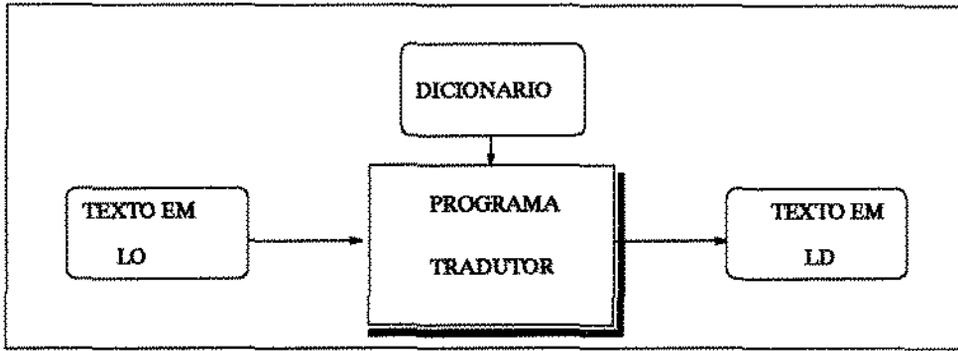


Figura 2.1: Sistema de tradução direto

um dicionário bilíngüe e a técnica de casamento de padrões podia ser produzido o texto na língua destino, sem uma análise rigorosa dos fenômenos lingüísticos que todo texto apresenta. Assim, com base nos conhecimentos sobre construções sintáticas típicas da *LO* e transformações das mesmas para a *LD* era obtida a tradução. Embora a qualidade obtida por esses sistemas fosse pobre, o documento resultante podia ser revisado por um tradutor humano, que produzia a versão final do documento [Hatim e Mason 1990].

Os sistemas de tradução mais modernos dividem o processamento em duas etapas principais, como pode ser observado na Figura 2.2. O texto na *LO* passa por um processo de entendimento que mapeia o texto em uma representação de seu significado. Essa representação pode ser utilizada por um sistema gerador de texto para produzir a tradução na *LD*.



Figura 2.2: Sistema de tradução baseado em entendimento

As representações de significado utilizadas por sistemas de tradução variam em função dos objetivos a serem atingidos [Danlos 1987]. Se o principal objetivo do sistema for a fidelidade da tradução, então uma representação “sintática” dos fenômenos pode ser suficiente, desde que existam regras de transformação entre estruturas sintáticas das línguas envolvidas no projeto. Entretanto, se o objetivo for a representação do conteúdo proposicional do texto, então representações essencialmente “semânticas” serão mais adequadas.

Podemos distinguir dois tipos de sistemas indiretos, de acordo com os processadores e com as representações utilizadas [Lewis 1992]: (i) sistemas *transfer*; e (ii) sistemas *interlíngua*.

Na Figura 2.3 é apresentada a arquitetura de um sistema transfer.

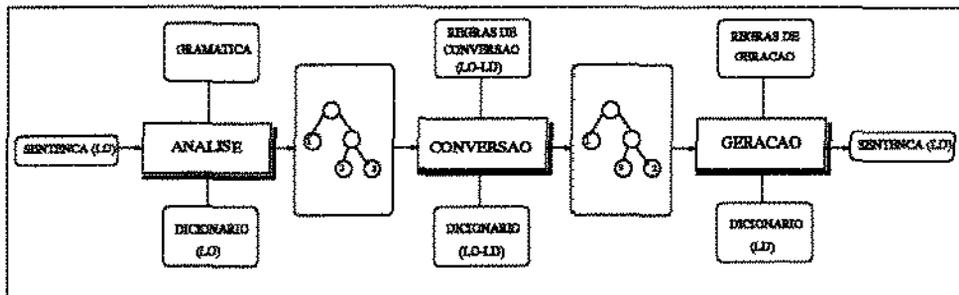


Figura 2.3: Sistema transfer

Um sistema transfer caracteriza-se por possuir: (i) um analisador para cada LO ; (ii) um gerador para cada LD ; e (iii) um processo de conversão para cada par $\langle LO, LD \rangle$. Um analisador em um sistema transfer apenas tem conhecimentos sobre a língua para a qual foi desenvolvido, desconhecendo para qual língua será realizada a tradução. A incorporação de uma nova LO requer a construção de um analisador e um conversor da nova língua para as outras LDs .

Na Figura 2.4 é apresentada a arquitetura de um sistema interlígua.

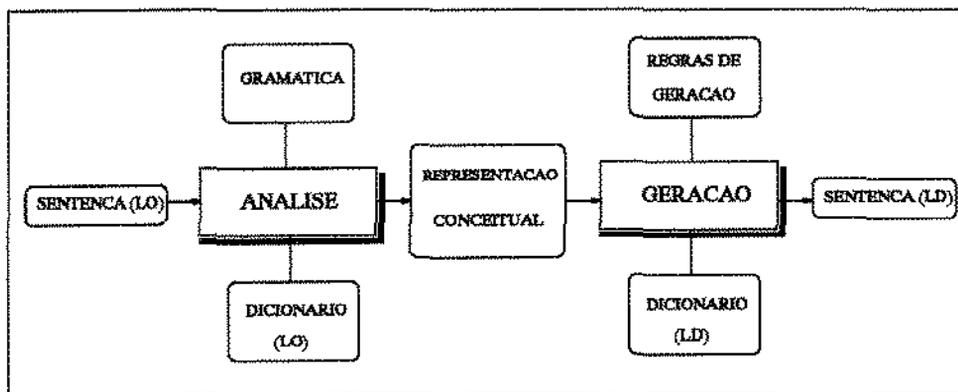


Figura 2.4: Sistema interlígua

Esta arquitetura caracteriza-se por possuir: (i) um analisador para cada LO ; e (ii) um gerador para cada LD . Neste paradigma o texto é processado e mapeado para uma representação chamada de interlígua, um formalismo de representação que deve incluir um conjunto de traços sintáticos e semânticos para possibilitar a representação abstrata de qualquer expressão das línguas envolvidas no sistema [Lewis 1992]. O desenvolvimento de um formalismo interlígua não é uma tarefa simples. Em geral o formalismo de representação será desenvolvido para cobrir os fenômenos lingüísticos das línguas envolvidas no projeto; portanto, a incorporação de uma nova língua implicará na adequação do formalismo, o que não é

trivial. Se a nova língua for uma *LO* deverá ser projetado um analisador; entretanto se a nova língua for uma *LD* deverá ser construído um gerador.

Na Inteligência Artificial, os paradigmas interlíngua têm sido utilizados extensamente para representar conceitualmente frases e textos. As representações também foram utilizadas na TA, mas as traduções obtidas nesse contexto resultavam em paráfrases do texto original. Isso se deve ao fato do formalismo não conter traços adequados de representação da estrutura do discurso.

Os sistemas de processamento de linguagem natural tem sido classificados em gerações segundo o fenômeno lingüístico abordado [Krusee 1991]. Sistemas de primeira geração tratam a palavra como principal fenômeno lingüístico e o processamento é muitas vezes feito utilizando-se a técnica de casamento de padrões; sistemas de segunda geração analisam a estrutura e o significado das sentenças, geralmente baseados em gramáticas e analisadores sintáticos; sistemas de terceira geração tratam o texto como um todo analisando relações entre sentenças e utilizando bases de conhecimento para o processamento textual. Temos notado que na maioria dos sistemas de tradução automática a sentença é a unidade fundamental. Um texto é considerado uma justaposição de sentenças e consoante com essa concepção a tradução é feita desconsiderando fenômenos que ultrapassam a barreira sentencial e que são próprios do fenômeno textual. Entretanto, se analisarmos a tarefa do tradutor sob o ângulo da lingüística [Hatim e Mason 1990], podemos constatar que uma das principais preocupações de um tradutor humano é a estrutura textual formada por segmentos e relações entre sentenças. O reconhecimento das relações entre as sentenças é fundamental para um tradutor, já que essas relações devem permanecer constantes após o passo de tradução.

Portanto, torna-se interessante a incorporação de um passo de análise textual em um sistema de tipo transfer para o tratamento de características próprias do fenômeno textual [Saggion e Carvalho 1995a]. Consideramos que um processo de cálculo da estrutura de um texto é importante para a obtenção de uma tradução de alta qualidade [Saggion e Carvalho 1995b]. Dada a complexidade que existe na análise de texto irrestrito decidimos restringir o domínio textual a sumários de artigos técnicos. Estamos interessados em definir um processo de análise de sumários em língua portuguesa como parte integrante da arquitetura apresentada na Figura 2.5.

Para isso, nesta dissertação duas questões devem ser abordadas: (i) a **estrutura de um sumário**; e (ii) o **processo de cálculo dessa estrutura**.

2.3 O domínio dos sumários

Um sumário é a primeira parte de um relatório técnico e apresenta um resumo dos tópicos a serem abordados no relatório. Segundo a Ciência da Documentação um

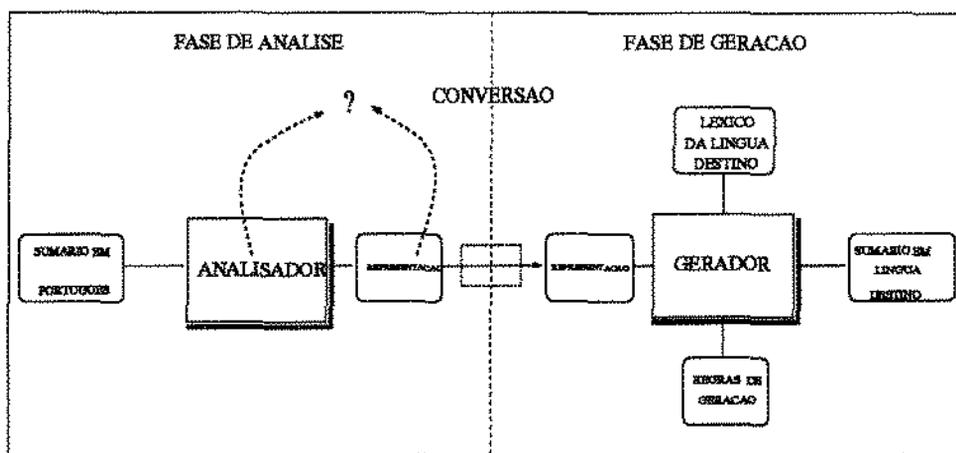


Figura 2.5: Sistema tradutor de sumários

sumário é um indicador do conteúdo do relatório (Figura 2.6) [Weil et al. 1963a, Cleveland e Cleveland 1983]. Segundo a Linguística [Hutchins 1985, Hutchins 1987, Gopnik 1972], apesar de muitas vezes ser restrito a um único parágrafo, é considerado um texto e apresenta todas as relações interessantes de textos maiores. Em particular nele aparecem relações de coesão e de coerência.

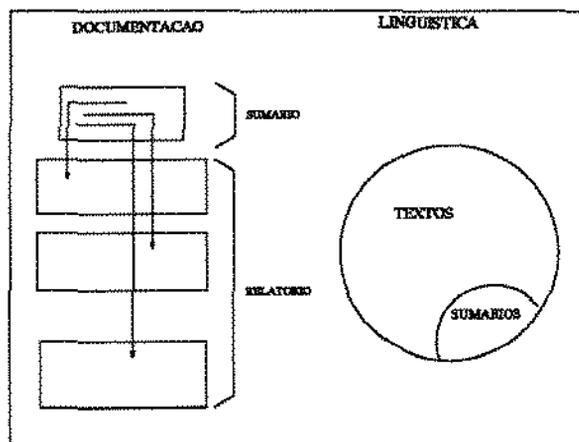


Figura 2.6: Os sumários: documentação e linguística

2.3.1 Tipos de sumários

A ciência da Documentação distingue entre dois tipos básicos de sumários:

- (i) Sumário *indicativo* ou *descritivo*.

É basicamente uma tabela de conteúdo do documento original, mas escrito em linguagem natural. Este tipo de sumário indica o material que será encontrado no relatório e jamais poderá transformar-se em um substituto deste último. Na Figura 2.7 podemos observar dois sumários indicativos. O primeiro apenas indica

o objetivo do artigo; o segundo indica o assunto do relatório.

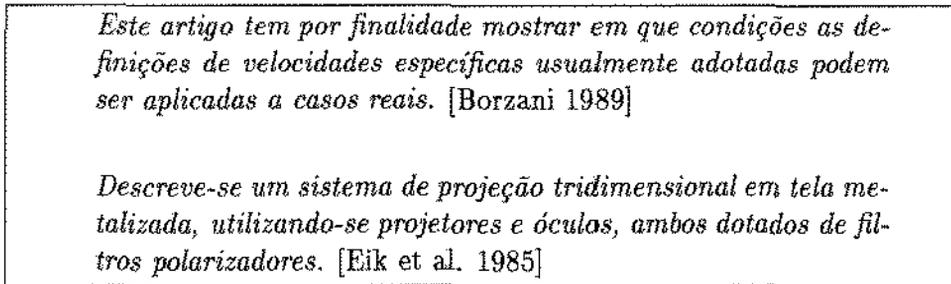


Figura 2.7: Sumários indicativos

(ii) Sumário *informativo*.

Trata-se de um verdadeiro texto, que apresenta os dados e as informações do relatório técnico, desenvolvidos de maneira lógica. Na Figura 2.8 podemos observar um sumário de tipo informativo que especifica os objetivos do artigo, explica o assunto e a metodologia de trabalho. Um sumário informativo pode em alguns casos transformar-se em substituto do documento. Na prática sumários são geralmente uma combinação dos dois tipos aqui considerados.

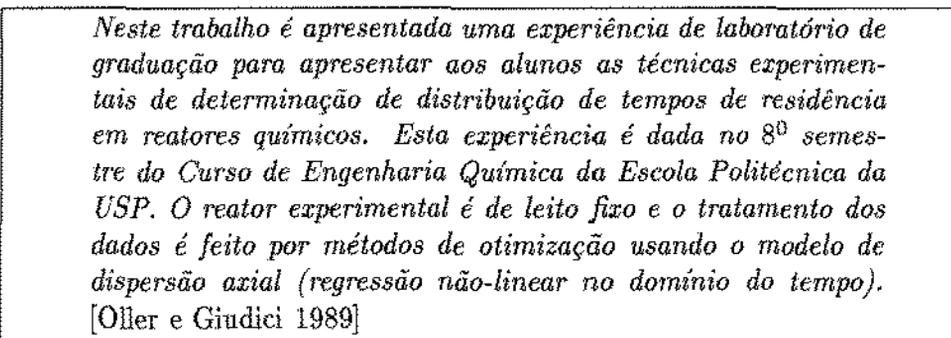


Figura 2.8: Sumário informativo

2.3.2 Estrutura de um sumário

Sumários dos campos de conhecimento mais diversos são escritos de maneira similar [Weissberg e Buker 1990, Weil et al. 1963b] e existem publicações que estabelecem guias sobre como este tipo de texto deve ser escrito, assim como o tipo de informação a ser incluída.

Um sumário incluirá em geral os seguintes tipos de informação:

- (i) *contexto*, onde informação preliminar é incluída e o problema estudado é definido ou identificado;

- (ii) *objetivos e escopo* do estudo apresentado no relatório;
- (iii) *metodologia* empregada na pesquisa, incluindo materiais, equipamentos e procedimentos utilizados na investigação;
- (iv) *resultados* obtidos na pesquisa;
- (v) *conclusões* alcançadas.

Além do tipo de informação que aparecerá em um sumário também podem ser verificadas normas de utilização dos tempos verbais. O tempo passado é utilizado para descrever informações relacionadas com o trabalho realizado. Entretanto o tempo presente é utilizado para descrever informações sobre o próprio relatório técnico.

2.3.3 Usos e aplicações dos sumários

A principal aplicação dos sumários é para documentação já que eles apresentam um resumo do tipo de material que pode ser encontrado no relatório técnico. Um sumário será geralmente escrito para reduzir o tempo e o esforço na busca de informação. Os sumários apresentam um panorama geral de um documento com as informações relevantes para que o leitor decida sobre a necessidade de consultar o documento original (Figura 2.9).

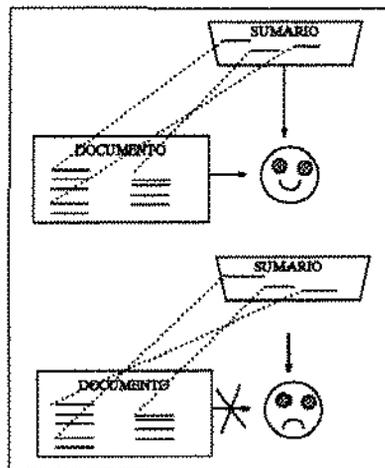


Figura 2.9: Os sumários e os leitores

Os sumários também servem para ultrapassar a barreira da língua; muitas publicações e congressos exigem a inclusão de um sumário em língua inglesa e, dado que a maioria dos pesquisadores compreende esta língua, os sumários servem para que o usuário decida sobre a conveniência de traduzir o relatório técnico completo para a sua língua. Alguns congressos decidem sobre a aceitação de um trabalho científico baseando-se no sumário apresentado pelo autor.

2.4 O corpus

O nosso trabalho baseia-se no estudo de um subconjunto de sumários de artigos técnicos publicados pela Revista de Ensino de Engenharia da Associação Brasileira de Ensino de Engenharia. Decidimos estudar esses textos devido ao fato dos sumários serem escritos respeitando as normas da Associação Brasileira de Normas Técnicas [ABNT 1987]. O estudo de sumários reais pareceu-nos a maneira apropriada para levar a término a pesquisa. A utilização de textos reais em lingüística computacional possibilita a observação de fenômenos que de outra maneira seriam simplificados. Além disso os aspectos teóricos por nós estudados podem ser verificados ou rejeitados na prática. Embora a escolha de um subconjunto de textos possa em princípio restringir a pesquisa, notamos que os fenômenos observados podem perfeitamente ser estendidos para outros domínios textuais.

Na Figura 2.10 apresentamos um sumário da publicação Revista de Ensino de Engenharia¹.

(1) Os programas de avaliação de docentes pelos alunos vêm sendo utilizados nos Estados Unidos desde a década de quarenta. (2) Este tipo de programa não é muito comum nas universidades brasileiras. (3) No Departamento de Engenharia Mecânica da UNICAMP foram feitas algumas tentativas de implantação destes programas. (4) O presente trabalho descreve a experiência da aplicação destes programas no Departamento de Engenharia Mecânica da UNICAMP. [Milanez 1987]

Figura 2.10: Um sumário do corpus (1)

Neste sumário podemos reconhecer os seguintes tipos de informação: nas sentenças (1) e (2) é apresentada informação de tipo contextual, visto que são introduzidas as entidades sobre as quais trata o artigo (os programas de avaliação); a sentença (3) contém informação sobre o trabalho científico desenvolvido (a implantação de programas de avaliação) e, finalmente, a sentença (4) apresenta os objetivos do relatório técnico (descrever a implantação dos programas de avaliação).

Na Figura 2.11 pode ser observado outro sumário do corpus analisado.

Neste texto podemos verificar a inclusão de informação de contexto. O problema estudado no relatório é descrito nas orações (1), (2) e (3); nas orações (4) e (5) são apresentadas respectivamente as sugestões (alterações na metodologia de ensino

¹Os números que aparecem no texto do sumário foram por nós colocados para referência nesta dissertação.

(1) Com a utilização de calculadoras programáveis e micro-computadores, (2) torna-se necessária a adequação do plano de ensino de Cálculo Numérico (3) para a formação dos futuros engenheiros. (4) Algumas alterações na metodologia de ensino desta disciplina são sugeridas e (5) feitas algumas recomendações quanto à utilização de calculadoras programáveis.
[de Araújo e Szeremeta 1985]

Figura 2.11: Um sumário do corpus (2)

de Cálculo Numérico) e as recomendações (uso de calculadoras programáveis).

2.5 Conclusões

Neste capítulo foram apresentados os conceitos básicos sobre tradução automática. Vimos que o processamento é realizado em duas etapas fundamentais: (i) análise do fenômeno lingüístico na língua origem; e (ii) síntese ou geração na língua destino. Em particular a etapa de análise apenas tem conhecimentos sobre a língua origem, desconsiderando-se por completo a língua destino para a qual será feita a tradução. Observamos também que muitas vezes os sistemas consideram a sentença e não o texto como unidade lingüística fundamental. Entretanto, o tratamento de fenômenos próprios dos textos e, em particular sua estrutura, pode trazer benefícios para a tradução automática.

Dada a complexidade do tratamento de um texto irrestrito decidimos restringir o domínio a sumários de artigos técnicos pois, embora curtos, são considerados textos com os mesmos fenômenos de interesse de textos maiores; além disso existe um interesse prático na tradução deste tipo de textos para outras línguas. Propusemos um sistema de análise de sumários em língua portuguesa como parte integrante de uma arquitetura transfer. Para especificar a arquitetura do analisador de sumários duas questões devem ser abordadas: (i) a estrutura dos sumários; e (ii) o processo de cálculo dessa estrutura.

Capítulo 3

Processamento Textual

Para processar adequadamente um texto é necessário considerar fenômenos que ultrapassam a barreira sentencial. Neste capítulo introduzimos dois conceitos tidos pela lingüística como fundamentais na definição de um texto: a *coesão* e a *coerência* textual.

3.1 A estrutura textual

Uma gramática procura capturar o conhecimento lingüístico que um falante tem sobre sua língua. Assim, os lingüistas tentam estabelecer um conjunto de princípios e regras através dos quais seja possível distinguir entre sentenças aceitáveis e não aceitáveis. Mas não existe, no momento, um formalismo que dê conta da estrutura lingüística do texto e que permita distinguir entre textos bem e mal formados [Hovy 1993]. Os estudos sobre textualidade tratam dos fenômenos observáveis nos textos e que diferem daqueles fenômenos observáveis nas sentenças.

Segundo [Halliday e Hasan 1976] um texto é uma unidade de sentido cuja manifestação superficial é uma seqüência de sentenças. Cabe então perguntarmos o que faz com que essa seqüência permaneça unida. Dois estudos são abordados para responder a essa pergunta: (i) a *coerência* textual; e (ii) a *coesão* textual [Lopes Fávero 1991].

De acordo com [Mateus et al. 1983] a coerência textual trata da conexidade conceitual. Um texto será coerente à medida que as relações entre as entidades e os eventos descritos no “mundo textual” sejam coerentes com as entidades e eventos do “mundo real”. O problema da coerência textual é transferido para o problema da coerência no mundo real.

A coesão textual [Koch 1989] estuda: (i) a referência; (ii) a co-referência; e (iii) a seqüenciação frásica. A coesão é uma relação semântica que se manifesta no sistema léxico-gramatical da língua [Halliday e Hasan 1976]; tipicamente cada

sentença na seqüência textual apresentará um vínculo coesivo com a sentença que a sucede ou a precede.

3.2 **Abordagens à estrutura textual**

A estrutura lingüística de um texto tem sido estudada de diferentes ângulos e com diferentes objetivos que incluem a geração de textos por computador e o entendimento de fenômenos da língua natural.

[McKeown 1985] estudou a estrutura de textos curtos produzidos em resposta a perguntas sobre a organização de um banco de dados, visando reproduzir esses padrões de produção textual em um sistema computacional de geração. Os estudos empíricos por ela realizados mostram que as pessoas seguem padrões estereotípicos na produção de textos. Mais ainda, ela nota que no discurso científico também os escritores utilizam o conhecimento sobre que tipo de informação deve ser colocada na introdução e a ênfase que deve ser dada na conclusão.

Textos curtos estudados por [Sidner 1978] revelam que as entidades descritas em um texto estão conceitualmente vinculadas ao mundo real. Ela tratou o problema da co-referência utilizando esse conceito e analisando as conexões entre sentenças adjacentes.

[Grosz e Sidner 1986] propuseram uma estrutura do discurso constituída de três componentes: (i) a estrutura lingüística; (ii) o estado atencional; e (iii) a estrutura intencional. O modelo computacional para processamento do discurso por elas proposto tem como principal objetivo a resolução da co-referência no discurso. A estrutura do discurso impõe restrições na escolha de um antecedente para um pronome.

[Hovy 1990] estudou as relações que vinculam as sentenças em um texto com a principal preocupação de produzir automaticamente textos coerentes. Vários autores na área de geração de textos utilizam representações nas quais as proposições estão vinculadas por relações retóricas ou de coerência. Ele propõe uma classificação hierárquica das relações em um texto coerente que compila os estudos de vários autores. Na sua classificação, as relações de coesão aparecem no topo da hierarquia e à medida que avançamos na classificação as relações tem características mais semânticas.

[Mann e Thompson 1983], baseados na análise de textos reais, definiram uma teoria que permite representar textos em função de relações que vinculam sentenças adjacentes e grupos de sentenças adjacentes na superfície lingüística. A teoria inclui relações tais como: circunstância, solução, elaboração, contraste e

causa. A verificação da existência de uma relação entre dois segmentos de textos, chamados respectivamente núcleo e satélite, depende: (i) do material contido nos segmentos; (ii) do contexto no qual o texto foi escrito; e (iii) das convenções culturais do escritor e dos potenciais leitores do texto.

O conjunto de relações especificadas dá conta de muitos tipos de textos em língua inglesa, mas não se trata de uma teoria fechada; ela pode ser estendida para outras línguas. A estrutura do texto segundo esta teoria é uma árvore na qual os nós internos são relações proposicionais e as folhas são proposições. Esta teoria foi utilizada na geração de textos por computador [Hovy 1993, Moore e Paris 1994].

3.3 A coerência textual

[Hatim e Mason 1990] consideram que um texto é coerente quando as sentenças elaboram continuamente uma entidade, apresentando-a nas suas diversas facetas conceituais. Assim, um *segmento de texto coerente* é aquele no qual uma entidade é continuamente elaborada. O limite de um segmento coerente pode ser determinado pela finalização dessa elaboração.

A Inteligência Artificial tem tido particular interesse na coerência textual. [Hobbs 1978a, Hobbs 1978b, Hobbs 1985] baseia a sua análise da coerência textual no descobrimento das relações que vinculam sentenças adjacentes. Para determinar essas relações é necessário um processo dedutivo e um banco de conhecimentos sobre como as entidades e os eventos estão relacionados no “mundo real”.

Segundo este autor o grau de coerência do texto dependerá da dificuldade em estabelecer uma relação de coerência na seqüência lingüística. Para calcular a vinculação entre as sentenças é necessário: (i) conhecimento do mundo; (ii) formalismos de representação desse conhecimento; (iii) operações sobre as representações; e (iv) um mecanismo de controle sobre as operações. As deduções serão feitas sobre as representações pela aplicação de operações controladas.

Assim, consideremos por exemplo o segmento de texto apresentado na Figura 3.1.

- | |
|---|
| (1) <i>Onde está Ariadne?</i>
(2) <i>Estou precisando de uma referência sobre anáfora.</i> |
|---|

Figura 3.1: Coerência em texto (1)

Para descobrir a coerência do segmento devemos: (i) traduzir as sentenças para um formalismo de representação; e (ii) realizar uma dedução no banco de conhecimentos para encontrar uma vinculação entre as representações obtidas. Suponhamos que as sentenças (1) e (2) sejam traduzidas respectivamente para as representações (3.1) e (3.2) da Figura 3.2.

$busca(eu, ariadne)$	(3.1)
$necessita(eu, ref(anafora))$	(3.2)

Figura 3.2: Representando sentenças (1)

Suponhamos também que o banco de conhecimentos contenha os fatos apresentados na Figura 3.3.

$sabe(ariadne, pln)$	(3.3)
$sabe(X, Y) \wedge tema(Z, Y) \Rightarrow sabe(X, Z)$	(3.4)
$tema(anafora, pln)$	(3.5)
$sabe(X, Y) \Rightarrow tem(X, ref(Y))$	(3.6)
$necessita(X, Z) \wedge tem(Y, Z) \Rightarrow busca(X, Y)$	(3.7)

Figura 3.3: Banco de conhecimentos (1)

Podemos estabelecer uma cadeia de deduções entre os fatos (3.1) e (3.2) utilizando o banco de conhecimentos e um mecanismo de dedução. Dessa maneira conseguimos vincular as sentenças (1) e (2) e concluir, então, que a seqüência é coerente no mundo representado.

[Hobbs 1985] identifica uma relação de coerência chamada *contraste* definida da seguinte maneira:

Sejam S0 e S1 duas asserções tais que a partir de S0 é possível inferir $p(a)$ e a partir de S1 é possível inferir $\sim p(b)$, onde a e b são "similares". Então dizemos que entre S0 e S1 existe uma relação de Contraste.

Como um exemplo consideremos o segmento de texto da Figura 3.4.

<p>(3) <i>Os programas de avaliação são comuns nas universidades americanas.</i></p> <p>(4) <i>Este tipo de programa não é comum nas universidades brasileiras.</i></p>

Figura 3.4: Relação de contraste em texto

A análise desse segmento produz duas proposições que podem ser observadas na Figura 3.5.

comum(programa_avaliação, universidade_americana)
 ~ *comum(programa_avaliação, universidade_brasileira)*

Figura 3.5: Representando sentenças (2)

Suponhamos que também temos o conhecimento apresentado na Figura 3.6.

similar(universidade_americana, universidade_brasileira)

Figura 3.6: Conhecimento necessário para a verificação da relação de contraste

Então podemos verificar a coerência do segmento utilizando essa relação.

[Mann e Thompson 1987] utilizam relações para verificar a coerência de um segmento de texto. A relação de *Elaboração* entre dois segmentos de texto N e S pode ser verificada se S apresentar detalhes adicionais sobre um elemento introduzido ou inferido de N a partir de uma relação conjunto-membro, abstrato-instância, todo-parte, processo-passo, objeto-atributo, geral-específico. Desta forma a seqüência da Figura 3.7 pode ser interpretada como coerente em virtude da relação de elaboração que se estabelece entre as sentenças. A elaboração é dada por uma relação todo-parte entre uma conferência e os artigos que serão apresentados na conferência.

- (5) *A conferência será realizada em São Paulo.*
 (6) *Os artigos serão recebidos até o dia...*

Figura 3.7: Relação de Elaboração em texto

Consideremos, como mais um exemplo de coerência, a Figura 3.8, onde apresentamos um segmento de história de “restaurante”.

Para resolver o problema da coerência desse segmento, [Schank e Abelson 1977] propuseram um formalismo de representação de conhecimento chamado de *script*. Basicamente o formalismo tenta compilar o conhecimento que uma pessoa tem sobre uma situação típica. Assim, um “script de restaurante” terá um conjunto de papéis que os atores irão desempenhar na história e seqüências de eventos típicos em histórias de restaurantes. Uma seqüência de sentenças será coerente neste modelo à medida que cadeias dedutivas possam ser estabelecidas entre os eventos descritos na seqüência lingüística e as cadeias de eventos no script. O processamento de um texto em um sistema de scripts é apresentado na Figura 3.9.

- (7) *João entrou no restaurante.*
 (8) *O garçom ofereceu-lhe o cardápio.*

Figura 3.8: Coerência em texto (2)

No decorrer da análise expressões são utilizadas para seleccionar um script que a

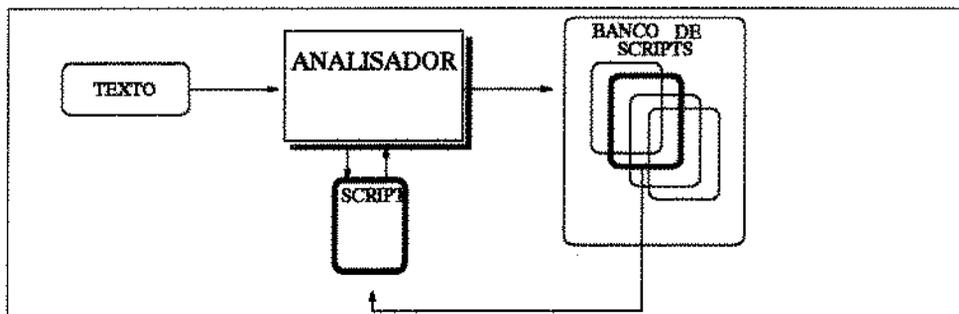


Figura 3.9: Scripts

partir da sua ativação atuará como contexto para o processamento do texto subsequente. Assim, no exemplo apresentado a expressão linguística “o restaurante” selecciona do banco de scripts o “script de restaurante” que vai atuar como contexto no entendimento da história toda. Uma vez que o script de restaurante foi seleccionado é possível identificar as entidades “o garçom” e “o cardápio”, típicas de histórias de restaurantes.

3.4 A coesão textual

Para que uma seqüência de sentenças possa ser chamada de texto devem existir vínculos entre as sentenças adjacentes. Para [Halliday e Hasan 1976] a coesão é o fator determinante da textualidade. Uma seqüência de sentenças será considerada um texto à medida que fenômenos coesivos sejam observados na superfície linguística. Devemos notar que para [Koch e Travaglia 1990], a coesão não é um fator nem necessário nem suficiente para que uma seqüência de sentenças possa ser chamada de texto; mas eles reconhecem que nos textos científicos esta é uma característica altamente desejável.

3.4.1 A coesão referencial

[Koch 1989] e outros pesquisadores distinguem dois tipos de coesão para o português: (i) a coesão referencial; e (ii) a coesão seqüencial. A coesão seqüencial estuda a seqüenciação frásica e a estrutura temática. Neste trabalho trataremos

apenas da coesão referencial, que passaremos a descrever.

Uma das propriedades que uma seqüência de sentenças deve ter para ser considerada um texto é a de assinalar, por meio de formas lingüísticas, se certos objetos e entidades estão sendo introduzidos pela primeira vez no discurso ou se já foram mencionados no discurso anterior.

A coesão referencial trata justamente desse problema e pode ser caracterizada mediante o gráfico que apresentamos na Figura 3.10.

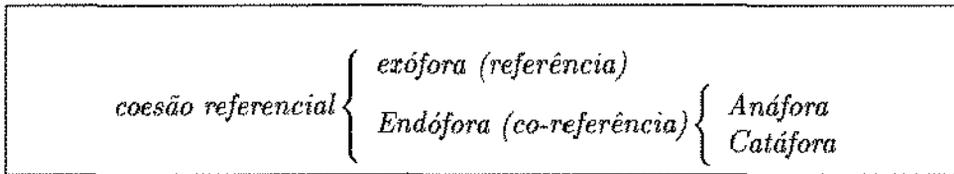


Figura 3.10: A coesão referencial

3.4.2 Exófora e endófora

Quando escrevemos utilizamos expressões lingüísticas para nos referir a entidades e situações do mundo real. O problema de como uma expressão lingüística se refere a uma entidade do mundo real ou conceitual é o clássico problema da referência semântica ou *exófora*. Por exemplo, na Figura 3.11 a expressão “os autores” refere-se a uma entidade fora do texto mas contida na situação comunicativa.

<p>(...) Os autores, após fazerem uma análise aprofundada da legislação existente sobre o assunto (...)</p>
--

Figura 3.11: Exófora

Entretanto, existem situações nas quais os escritores utilizam *formas abreviadas* para se referir a uma entidade do próprio texto [Hirst 1981]; este fenômeno é denominado *endófora*. A forma abreviada é chamada de *item remissivo* e a expressão a qual o item remissivo se refere chama-se *antecedente*. O processo pelo qual determina-se o antecedente de um item remissivo chama-se *resolução*. Quando se estabelece uma relação deste tipo entre o item remissivo e o antecedente dizemos que ambos são *co-referentes* ou que estão em uma relação de

co-referência.

3.4.3 Anáfora e catáfora

Dois tipos de co-referência são identificados: (i) quando o item remissivo aparece antes do seu antecedente na seqüência lingüística estamos em uma situação de *catáfora*; (ii) quando o item remissivo aparece depois do seu antecedente na seqüência lingüística estamos em uma situação de *anáfora*.

Na Figura 3.12 podemos observar uma referência de tipo anafórica. A expressão lingüística “elas” é interpretada como co-referente com a expressão lingüística “as ciências”, introduzida previamente no texto. O item remissivo é o pronome “elas” e o antecedente é a expressão “as ciências”.

A engenharia depende cada vez mais das ciências e de técnicas nelas baseadas.

Figura 3.12: Anáfora

Na Figura 3.13 a expressão lingüística “ele” é interpretada como co-referente com a expressão “João” introduzida posteriormente ao item remissivo.

Ele quebrou o carro! Foi João quem quebrou o carro!

Figura 3.13: Catáfora

Em resumo, consideramos itens remissivos aquelas expressões da língua que não podem ser interpretadas semanticamente por si mesmas, mas que remetem a outros itens do discurso, necessários à sua interpretação.

Se considerarmos o segmento de texto da Figura 3.14 notaremos que a expressão “a disciplina Similitude em Engenharia” da sentença (9) tem um referente particular e concreto no domínio discursivo.

(9) *O presente trabalho tem como meta divulgar a disciplina Similitude em Engenharia.*
 (10) *Esta disciplina visa proporcionar aos alunos fundamentos básicos da teoria de modelos.*

Figura 3.14: Referência e co-referência (1)

Entretanto, uma expressão lingüística como “esta disciplina” não tem referência própria pois depende do contexto no qual ela é utilizada. No caso particular da sentença (10) ela é interpretada pela sua relação com a expressão “a disciplina Similitude em Engenharia” introduzida previamente no discurso. Essa relação de coesão textual faz com que a seqüência lingüística seja interpretada como uma unidade textual.

3.4.4 Anáfora profunda

Considere o texto da Figura 3.15.

Na sala havia apenas uma escrivaninha. A escrivaninha era velha, e as gavetas estavam emperradas.

Figura 3.15: Anáfora profunda

Na Figura 3.16 apresentamos o esquema de referência e co-referência textual deste texto.

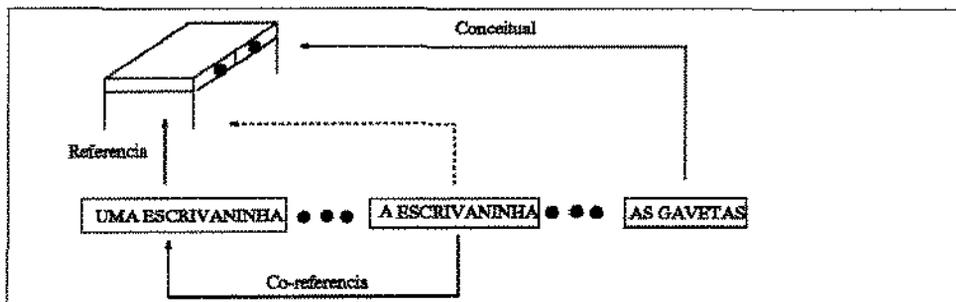


Figura 3.16: Referência e co-referência (2)

A expressão lingüística “uma escrivaninha” contém instruções de sentido para a criação de um referente *escrivaninha* que tem todas as propriedades de uma escrivaninha. A expressão lingüística “a escrivaninha” se refere a mesma entidade *escrivaninha* em virtude da relação que essa expressão tem com a expressão “uma escrivaninha”. As duas expressões consideradas são *co-referentes*. A expressão lingüística “as gavetas” obtém o seu referente em virtude da relação conceitual que mantém com a expressão “uma escrivaninha”. O fato de uma *escrivaninha* ter geralmente *gavetas* possibilita a obtenção do referente dessa expressão lingüística. Este tipo de relação coesiva é muitas vezes chamada de anáfora semântica ou anáfora profunda.

3.5 Formas abreviadas no português

3.5.1 Elipse

No português existem várias formas de referência abreviada. *Elipse* é a ocorrência de um elemento sem realização lexical na seqüência lingüística. Na oração (12) da sentença da Figura 3.17 observamos um caso de sujeito elidido.

(11) *João caiu* e (12) *machucou a perna.*

Figura 3.17: Elipse

O sujeito elidido é interpretado como co-referente com o sintagma nominal “João” da oração (11). Para determinar o antecedente de um elemento elidido devem ser levados em consideração aspectos tais como: concordância de traços sintáticos, e estrutura sintática da frase imediatamente anterior. Em geral o elemento elidido pode ser interpretado como co-referente com um elemento da sentença ou oração precedente que tem a mesma função gramatical do elemento elidido. Assim, no exemplo, o elemento elidido é o sujeito da oração e seu antecedente é o sujeito da oração anterior.

3.5.2 Pronomes

O segundo elemento abreviado considerado são os *pronomes de terceira pessoa* que classificamos como:

- (i) Pronomes Pessoais = { ele, ela, eles, elas }
- (ii) Pronomes Objetivos = { o, a, os, as }
- (iii) Pronomes Reflexivos = { se, si próprio, si própria }
- (iv) Pronomes Demonstrativos (I) = { este(s), esta(s), esse(s), essa(s), aquele(s), aquela(s) }
- (v) Pronomes Demonstrativos (II) = { isto, isso, aquilo }

Estas expressões lingüísticas são chamadas de formas livres pois apenas trazem instrução de conexão com o seu antecedente considerando os traços de *gênero*, *número* e *pessoa*. A seguinte restrição imposta pela gramática é em geral considerada para a obtenção de um antecedente para um pronome:

- (I) Um pronome deve concordar nos traços de *gênero*, *número* e *pessoa* com o seu antecedente.

Os pronomes demonstrativos de tipo (I), além de trazerem instruções de conexão, trazem instruções de localização do antecedente. Assim, por exemplo na seqüência textual apresentada na Figura 3.18, o pronome “este” da sentença (14) é co-referente com “João” da sentença (13) e o pronome “aquele” da sentença (15) é co-referente com “Pedro” da sentença (13).

- | |
|---|
| <p>(13) <i>João e Pedro foram ao cinema.</i>
 (14) Este <i>estava de carro.</i>
 (15) Aquele <i>estava de moto.</i></p> |
|---|

Figura 3.18: Pronomes demonstrativos

Já os pronomes demonstrativos de tipo (II) remetem em geral a fragmentos oracionais, enunciados e contextos.

3.5.3 Frase nominal definida

O terceiro elemento considerado como uma referência abreviada é a *frase nominal definida* (FND) que em algumas ocasiões se refere à informação precedente. Uma FND é um sintagma nominal com operações de *determinação*. Essas operações de determinação estão refletidas no uso dos *determinantes* do português:

- (i) Artigos Determinados = { o(s), a(s) }; e
- (ii) Artigos Demonstrativos = { este(s), esta(s), esse(s), essa(s), aquele(s), aquela(s) }.

As frases nominais definidas têm, além de instruções de concordância, instruções de sentido, já que restringem o campo semântico do antecedente a ser localizado utilizando informação do predicador associado à frase. Esse predicador indica o tipo de material a ser localizado. As operações de determinação fazem com que seja necessária a busca de um referente no contexto ou no texto. Quando um escritor utiliza uma FND ele está pressupondo a capacidade do leitor de reconhecer a entidade a qual a frase se refere.

Consideremos por exemplo o segmento de texto da Figura 3.19.

A frase nominal definida “o método” da sentença (17) é uma referência abreviada à entidade representada pela expressão “um método simples” na sentença (16); notamos a co-referência pelo uso do mesmo núcleo em ambas as expressões.

Entretanto, nem sempre é utilizado o mesmo predicado nominal para estabelecer a conexão; ela pode ser estabelecida pelo uso de expressões sinônimas ou quase sinônimas. Consideremos o segmento apresentado na Figura 3.20 onde a

- (16) *O trabalho apresenta um método simples.*
 (17) *O método é usado para testar trocadores de calor.*

Figura 3.19: Anáfora definida em texto (1)

expressão “a moça” da sentença (19) é co-referente com a expressão “uma garota” da sentença (18).

- (18) *Uma garota veio lhe procurar quando você estava fora.*
 (19) *A moça disse que voltará logo.*

Figura 3.20: Anáfora definida em texto (2)

Embora o núcleo do sintagma nominal utilizado nas expressões difira, existe co-referência devido ao fato das expressões pertencerem ao mesmo campo semântico. O fato de uma “garota” ser considerada uma “moça” possibilita o estabelecimento da vinculação.

3.5.4 Outras formas abreviadas na língua portuguesa

[Koch 1989] considera ainda como formas abreviadas anafóricas os seguintes elementos da língua portuguesa:

- (i) Os pronomes possessivos = { meu, teu, seu, nosso, vosso, dele } e seus respectivos femininos e plurais.
 No exemplo (a) da Figura 3.21 o item “meu” refere-se ao falante.
- (ii) Os pronomes indefinidos = { todos, tudo, nenhum, vários }
 No exemplo (b) da Figura 3.21 o item “todos” refere-se a “João, Maria e Pedro”.
- (iii) Os pronomes interrogativos = { que, quanto, qual } e seus respectivos femininos e plurais.
 No exemplo (c) da Figura 3.21 o item “qual” refere-se ao maior país da América Latina.
- (iv) Os pronomes relativos = { que, o qual, quem }
 No exemplo (d) da Figura 3.21 a expressão “as quais” refere-se a “as cem pessoas”.
- (v) Os cardinais = { um, dois, três, ... }
 No exemplo (e) da Figura 3.21 a expressão “os dois” refere-se a “Pedro e João”.

- (i) **meu** carro está na oficina.
- (ii) João gosta de feijoada, Maria não perde o carnaval no Rio, Pedro sempre bebe caipirinha. **Todos** são brasileiros.
- (iii) **Qual** é o maior país da América Latina?
- (iv) Há cem pessoas lá fora **as quais** querem falar com o senhor governador.
- (v) Pedro e João foram ao cinema. **Os dois** gostaram do filme.
- (vi) Maria, Carla e Angela são irmãs. **A primeira** estuda computação, **a segunda** estuda teatro e **a última** pratica esportes.
- (vii) João comprou uma dúzia de bananas e deu-me a **metade**.
- (viii) Vi a Maria sentada no parque hoje de manhã. Quando voltei do banco ainda estava **lá**.
- (ix) A engenharia depende da criatividade. **Pensando dessa forma** estamos desenvolvendo uma experiência criativa na escola de engenharia.

Figura 3.21: Outras formas de referência abreviada

- (vi) Os ordinais = { primeiro, segundo, ..., último }
 No exemplo (f) da Figura 3.21 a expressão “a primeira” refere-se a “Maria”, a expressão “a segunda” refere-se a “Carla” e a expressão “a última” refere-se a “Angela”.
- (vii) Os fracionários = { metade, terço, ... }
 No exemplo (g) da Figura 3.21 a expressão “a metade” refere-se a “as bananas”.
- (viii) Os advérbios pronominais = { lá, aí, ali, aqui, onde }
 No exemplo (h) da Figura 3.21 o item “lá” refere-se a “o parque”.
- (ix) Expressões adverbiais = { assim, desse modo, ... }
 No exemplo (i) da Figura 3.21 a expressão “essa forma” refere-se à frase imediatamente anterior.

3.6 Restrições sintáticas à referência pronominal

Além das restrições nos traços de gênero, número e pessoa, existem outras restrições para a determinação dos possíveis referentes de expressões pronominais

tanto no domínio sentencial como no domínio textual, dependendo da teoria adotada.

Assim, por exemplo, na teoria gerativa do português [Lobato 1986] existe um conjunto de princípios que tornam possível a interpretação dos pronomes no nível sentencial. Consideremos por exemplo a sentença (20) da Figura 3.22 e a sua representação em termos de categorias sintáticas¹.

<p>(20) <i>a engenharia depende das ciências e de [técnicas</i> <i>n[elas]_{SN₂} baseadas]_{SN₁}</i></p> <p>(21) <i>a engenharia não prescinde d[a criatividade de quem</i> <i>[a]_{SN₂} usa]_{SN₁}</i></p>

Figura 3.22: Pronomes e categorias sintáticas

O pronome pessoal “elas” (SN_2) não pode ter como antecedente o sintagma nominal completo cujo núcleo é “técnicas” (SN_1) embora eles concordem em gênero (feminino) e número (plural). Este caso pode ser explicado por um princípio da gramática chamado de *Filtro de i sobre i* [Lobato 1986, Raposo 1992] pelo qual uma expressão nominal B não pode ser co-referente com uma expressão nominal A na qual B está contida.

Outro caso da mesma restrição é verificado na sentença (21) da mesma Figura 3.22, onde o pronome objeto “a” (SN_2) não pode ter como antecedente o sintagma nominal cujo núcleo é “criatividade” (SN_1).

Essa teoria impõe outras restrições considerando três noções fundamentais para o tratamento das relações de co-referência. Essas restrições baseiam-se em relações entres os nós de uma árvore e incluem a definição de *Dominância* em árvores e as definições específicas de *Regência*, *C-comando* e *Categoria de Regência* [Lobato 1986]. Estas tres últimas baseiam-se em árvores de análise sintática e em categorias sintáticas.

Definição 1 *Dominância:*

Um nó α domina um nó β se e somente se:

- (i) existir uma seqüência conexa de um ou mais ramos entre α e β ; e*
- (ii) o percurso de α até β através desses ramos for unicamente descendente.*

Definição 2 *C-comando:*

Um nó α c-comanda um nó β se e somente se:

¹As categorias são apresentadas no Capítulo 4.

- (i) α não domina β ;
- (ii) β não domina α ;
- (iii) a primeira categoria ramificante que domina α também domina β .

Uma categoria α se diz ramificante se α tem pelo menos dois nós filhos.

Quando um nó α c-comanda um nó β diz-se então que β está no domínio de α .

Definição 3 *Regência:*

Um nó α rege um nó β se e somente se:

- (i) α é uma categoria lexical² V, A, N ou P ou a categoria lingüística flexão $FLEX$;
- (ii) α c-comanda β e β não está protegido de α por uma das categorias SV, SA, SN ou SP .

Um nó β está protegido de α pela categoria X se X inclui β e X não inclui α .

Definição 4 *Categoria de regência:*

A categoria de regência de β é o SN ou S mínimo que inclui:

- (i) β ; e
- (ii) o regente de β .

Definidas estas relações sintáticas a teoria estabelece as seguintes condições sobre a co-referência:

- (II) Um pronome deve estar livre na sua categoria de regência.

Em outras palavras o antecedente para um pronome não pode c-comandar o pronome na sua categoria de regência.

- (III) Um reflexivo deve estar ligado na sua categoria de regência.

Em outras palavras um reflexivo deve estar c-comandado pelo seu antecedente na sua categoria de regência.

Assim, por exemplo, na sentença (22) da Figura 3.23, o pronome objeto “o” (SN_3) não pode ter como antecedente o sintagma nominal completo “o pai do João” (SN_1) já que a teoria torna válidas as seguintes afirmações:

²As categorias são apresentadas no Capítulo 4.

- (i) S é a Categoria de Regência do pronome;
- (ii) $SN_1 \in S$; e
- (ii) SN_1 c-comanda SN_3 .

Entretanto, o sintagma nominal “o João” (SN_2) pode ser co-referente com o pronome “o” (SN_3) pois embora “o João” (SN_2) pertença à categoria de regência do pronome “o” (SN_3), SN_2 não c-comanda SN_3 .

Na Figura 3.24 observamos a relação de c-comando entre SN_1 e SN_3 . Esta teoria gramatical impõe restrições de *exclusão* para a interpretação de antecedentes para pronomes não reflexivos.

(22) [[O pai d[o João] $_{SN_2}$] $_{SN_1}$ viu-[o] $_{SN_3}$ no parque] $_S$

(23) [[Maria] $_{SN_1}$ viu-[se] $_{SN_2}$ no espelho] $_S$

Figura 3.23: Restrições no nível sentencial

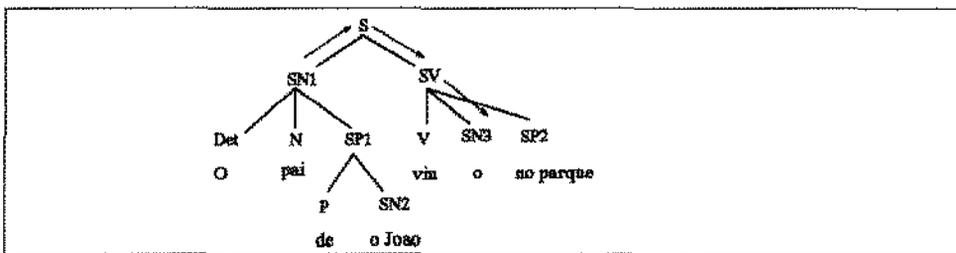


Figura 3.24: C-comando (1)

Já na sentença (23) da mesma Figura 3.23, o pronome reflexivo “se” (SN_2) tem como antecedente o sintagma nominal “Maria” (SN_1) devido aos seguintes fatos:

- (i) S é a Categoria de Regência de SN_2 ;
- (ii) $SN_1 \in S$; e
- (iii) SN_1 c-comanda SN_2 .

Na Figura 3.25 observamos a relação de c-comando entre os sintagmas. A teoria estabelece então quais devem ser os antecedentes para itens reflexivos.

3.7 Tratamento computacional de referência e co-referência

Todos os itens nominais da língua devem passar por um processo de interpretação de modo a obter ou criar o seu referente pois, são os nominais as categorias

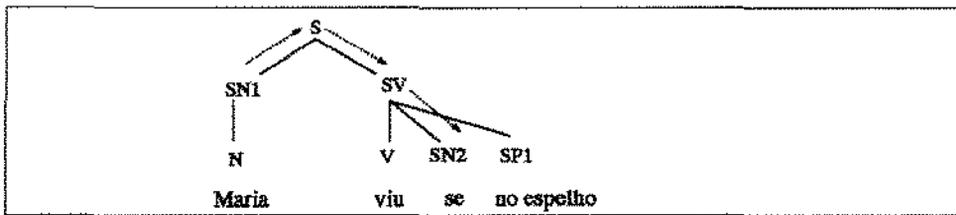


Figura 3.25: C-comando (2)

lingüísticas caracterizáveis por terem potencial de referência. Os nomes têm no léxico da língua instruções de sentido para construir o seu referente. Entretanto os pronomes apenas trazem instruções de conexão e carecem de identidade própria [Raposo 1992].

[Wilks 1973a, Wilks 1973b] foi um dos primeiros a atacar o problema da resolução de pronomes no domínio da tradução automática. Ele define uma teoria conhecida como *semântica preferencial* que possibilita a interpretação de pronomes na forma “preferida” ou mais natural. As formas não naturais de interpretação são consideradas como metafóricas.

Considere a seqüência de sentenças da Figura 3.26.

- (24) *Give the bananas to the monkeys.*
 (25) *Althoght they₁ are not ripe they₂ are hungry.*

Figura 3.26: Pronomes e tradução automática

Se o objetivo for traduzir essa seqüência para a língua portuguesa, obrigatoriamente deveremos obter os antecedentes dos pronomes “they₁” e “they₂” da sentença (25), pois o português diferencia os pronomes plurais de terceira pessoa no traço de gênero. Portanto, a tradução do pronome “they” do inglês dá lugar a duas possíveis alternativas no português: “eles” e “elas”.

Notamos, entretanto, que a seqüência apresentada tem uma leitura “preferida” que é a seguinte: a expressão “they₁” é interpretada como co-referente com a expressão “the bananas” da sentença (24) e a expressão “they₂” é interpretada como co-referente com a expressão “the monkeys” da sentença (24).

Para resolver as referências na forma preferida utiliza-se informação sobre os tipos semânticos dos argumentos dos verbos. No caso apresentado na Figura 3.26 podemos supor que as expressões “they₁ are not ripe” e “they₂ are hungry” são traduzidas, respectivamente, para as representações *not(ripe(they₂))* e *hunger(they₁)*. Como o predicador *ripe* exige um argumento de tipo *fruta*, a expressão “they₁”

herda esse traço semântico e então a expressão resulta co-referente com “the bananas”. Já o predicador *hunger* exige um argumento de tipo *animado*; esse traço semântico é herdado pela expressão “they₂” e desse modo essa expressão resulta co-referente com “the monkeys”. A dificuldade desta abordagem é que exige conhecimento detalhado do mundo.

Para a resolução de pronomes no domínio sentencial [Hobbs 1978c] utiliza restrições sintáticas aplicadas à árvore de análise. O antecedente para um pronome será uma frase nominal que **precede** e **não domina** o pronome na estrutura da árvore e que concorda em gênero e número com o pronome. No domínio intersentencial, o algoritmo considera todas as frases analisadas até então, utilizando a estratégia **última analisada - primeira examinada**. Cada árvore é percorrida em largura primeiro e depois em profundidade; toda frase nominal encontrada é considerada um antecedente para o pronome desde que não sejam violadas as condições de concordância sintática.

[Allen 1987] resolve as referências abreviadas e não abreviadas em um texto, apresentando um algoritmo baseado em uma estrutura de memória de lista que contém os nominais introduzidos no discurso anteriormente à ocorrência da anáfora; cada entrada na lista deve conter informações de tipo semântica e sintática para possibilitar referências posteriores. O antecedente para um pronome é buscado na lista considerando os traços de gênero e número associados ao pronome.

Para a obtenção dos referentes dos outros sintagmas nominais utiliza-se o seguinte critério:

- (i) Para uma frase indefinida é criado um novo referente;
- (ii) Para um nome próprio busca-se o referente em uma tabela; portanto, os nomes próprios sempre encontrarão o mesmo referente;
- (iii) Uma frase nominal definida é considerada anafórica; portanto, o antecedente será buscado na lista de entidades considerando informações semânticas.

Em [Grosz e Sidner 1986] o próprio discurso restringe a escolha dos antecedentes para um pronome. A estrutura do discurso é formada por segmentos e sub-segmentos. Quando um sub-segmento é finalizado, os referentes nele introduzidos ficam inacessíveis para os sub-segmentos que se seguem. O processamento é realizado utilizando-se uma pilha, onde o topo da pilha contém os referentes acessíveis ao segmento atual. Quando o processamento de um segmento é finalizado os referentes são desempilhados e então ficam inacessíveis. Esse tipo de mecanismo poderia ser aplicado ao texto apresentado na Figura 3.27, onde podemos observar dois segmentos encaixados S1 e S2.

A frase “fazendo um parênteses” é um indicador de começo de um sub-segmento (S2) que finaliza quando é encontrada a frase “agora voltando ao nosso assunto”.

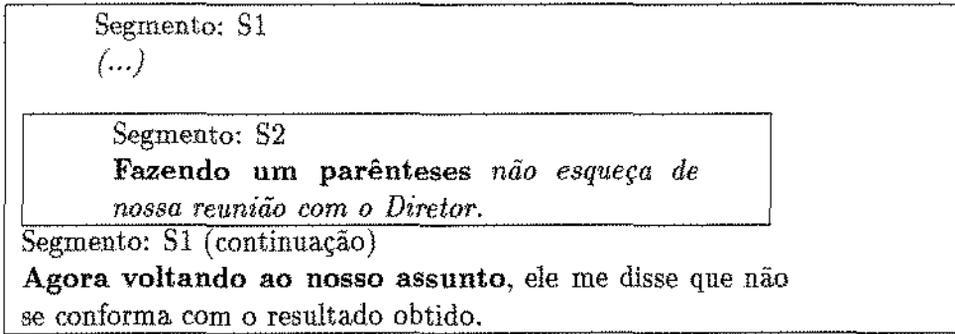


Figura 3.27: Segmentos e sub-segmentos

Esta última frase faz com que os referentes introduzidos no segmento S2 não possam ser utilizados como antecedentes da expressão pronominal “ele” do segmento S1.

3.7.1 Teorias sintáticas na resolução de pronomes

A utilização de teorias sintáticas é essencial para a resolução de pronomes em tradução automática [Saggion e Carvalho 1994]. Consideremos novamente a sentença (23) da Figura 3.23; o pronome “se” pode ser utilizado para se referir a entidades de gênero masculino, feminino e neutro. Por isso a tradução do pronome para a língua inglesa pode corresponder a qualquer uma das formas “himself”, “herself” ou “itself”; entretanto notamos que o pronome “se” tem como único possível antecedente o sintagma nominal “Maria” obtido a partir das restrições impostas pela teoria gramatical (seção 3.6). O pronome “se” herda dessa maneira o traço sintático feminino, de maneira a resolver o problema da ambigüidade. O pronome “se”, neste caso, será traduzido para a língua inglesa como “herself”.

Um processo computacional de resolução de pronomes pode ser observado na Figura 3.28.

Na resolução de pronomes dois estágios são considerados: (i) cálculo das entidades que não podem ser antecedentes para o pronome (restrições estruturais); e (ii) busca pelo antecedente do pronome considerando as entidades prévias [Saggion e Carvalho 1995c]. As anáforas reflexivas e recíprocas devem ser resolvidas utilizando-se apenas restrições estruturais.

No passo (i) deve ser obtido o conjunto de referentes que não podem ser antecedentes para o pronome; neste processo devem ser consideradas apenas restrições estruturais especificadas na teoria gramatical; por isso uma das informações a serem utilizadas neste estágio é a árvore de análise da sentença. Para cada anáfora i_a é calculado o conjunto $NC(i_a)$ de entidades não co-referenciais com i_a (3.8).

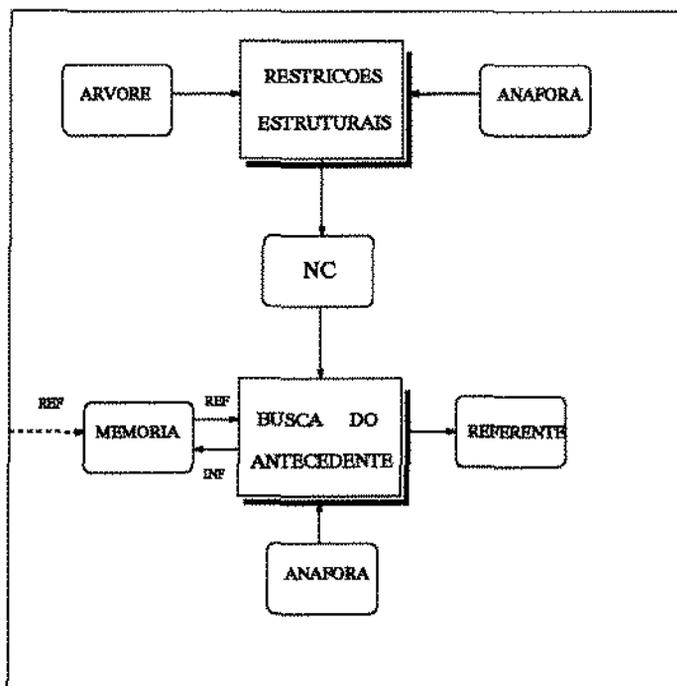


Figura 3.28: Resolução dos pronomes

$$NC(i_a) \equiv \{i_{a_1}, \dots, i_{a_k}\} \quad (3.8)$$

tal que i_{a_j} ($1 \leq j \leq k$) não é co-referente com i_a

No passo (ii) deve ser consultada uma estrutura de memória contendo os referentes introduzidos no discurso prévio. A memória é atualizada pelos referentes obtidos a partir das análises sintática e semântica processados previamente à ocorrência da anáfora; a cada referente devem ser associadas informações sintáticas e semânticas de maneira a tornar possível a sua localização. A busca na memória é realizada considerando-se informações sobre gênero, número e traços semânticos obtidos a partir da análise sintático-semântica da anáfora. Os referentes obtidos da memória a partir dessas informações e que não pertencem ao conjunto $NC(i_a)$ são possíveis antecedentes para a anáfora.

3.7.2 Necessidade de resolução de frases nominais definidas

Não só os pronomes mas também as frases nominais definidas tem que ser resolvidas em sistemas de tradução automática por vários motivos. Considere o texto apresentado na Figura 3.29.

A frase nominal definida “the singer” da sentença (27) deverá ser resolvida tendo como antecedente o sintagma nominal “Madonna”, para obter uma correta tradução da expressão “the singer” para o português, língua que diferencia entre

- (26) *Madonna met Alan Parker last night.*
 (27) *He tried to convince **the singer** to play the principal role in the film "Evita".*

Figura 3.29: Frases definidas e tradução automática

as expressões “o cantor” (masculino) e “a cantora” (feminino), ambas possíveis traduções da expressão “the singer”.

Além do problema da tradução automática, todo sistema de entendimento de linguagem natural deve estabelecer a conexão entre as frases nominais definidas e os seus antecedentes; caso contrário o sistema será incapaz de demonstrar um comportamento “inteligente”.

Considere o segmento de texto apresentado na Figura 3.30. Encontrar o antecedente da frase nominal definida “o presidente” da sentença (29) é fundamental para poder responder a uma pergunta como a que apresentamos na Figura 3.31.

- (28) *FHC^a assistiu ontem ao concerto no Municipal do Rio.*
 (29) ***O presidente** estava acompanhado por Dona Ruth Cardoso.*

^aPresidente do Brasil

Figura 3.30: Frases definidas e entendimento (1)

Quem acompanhava FHC no Municipal do Rio?

Figura 3.31: Frases definidas e entendimento (2)

Para resolver este tipo referência é necessário utilizar informação sobre as propriedades das entidades no domínio discursivo. Assim, por exemplo, o fato de “FHC” ser presidente e o fato de “Madonna” ser uma cantora são levados em consideração na resolução desse tipo de referência.

Em [Winograd 1973] um programa simula o comportamento “inteligente” de um robô que atua sobre um mundo de blocos como o apresentado na Figura 3.32. O robô é capaz de manipular os objetos do mundo para modificar o seu estado e para responder a perguntas sobre o estado atual. O usuário comunica-se com o robô utilizando uma interface em linguagem natural. O robô é capaz de entender as sentenças produzidas pelo usuário, ou seja é capaz de identificar os referen-

tes dos sintagmas nominais e as ações que ele deve realizar. A identificação dos sintagmas nominais com objetos do mundo é possível graças à representação por extenso dos elementos e de suas propriedades.

Consideremos o mundo de blocos da Figura 3.32 e a representação desse mundo em um formalismo lógico como o apresentado na Figura 3.33. Utilizamos o predicado *bloco*(*X*) para indicar que *X* é um bloco; utilizamos o predicado *azul*(*X*) para indicar que *X* é azul; utilizamos o predicado *branco*(*X*) para indicar que *X* é branco e o predicado *acima - de*(*X*, *Y*) para indicar que *X* está acima de *Y*.

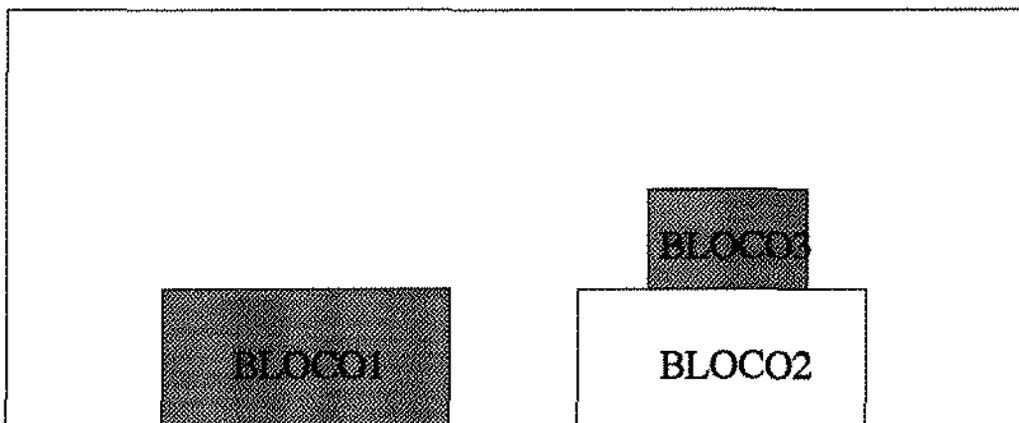


Figura 3.32: Um mundo de blocos

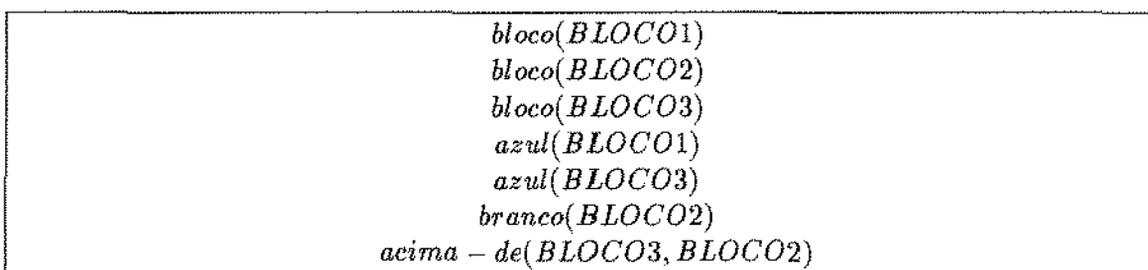


Figura 3.33: Representação do mundo dos blocos

Suponhamos ainda que a expressão em língua natural “o bloco azul” seja mapeada na representação (3.9).

$$\textit{azul}(X) \& \textit{bloco}(X) \tag{3.9}$$

O processo de interpretação da expressão deve instanciar o valor da variável *X* para obter o referente da expressão no mundo dos blocos. Note, no entanto, que existem duas possíveis soluções para a variável *X* no mundo da Figura 3.33. Isso significa que uma expressão como “pegue o bloco azul” resulta ambígua para o robô. Entretanto a expressão “o bloco azul acima do bloco branco” cujo mapeamento para a linguagem de representação é a expressão (3.10)

$$\text{azul}(X) \& \text{bloco}(X) \& \text{azul}(Y) \& \text{bloco}(Y) \& \text{acima} - \text{de}(X, Y) \quad (3.10)$$

resulta não ambígua no mundo representado.

3.8 A coesão referencial no corpus

No corpus de sumários analisados vários são os mecanismos de coesão referencial utilizados. Porém, o principal elemento para o estabelecimento de relações de coesão nos textos é a utilização de frases nominais definidas. Entretanto, um dos principais problemas detectados durante a fase de análise do corpus foi a observação de que nem toda frase nominal definida se refere a informação precedente. Assim, por exemplo na Figura 3.12 a expressão definida “a engenharia” não remete a nenhuma informação precedente. Notamos que essa expressão sempre se refere à mesma entidade independente do contexto na qual ela é utilizada. O mesmo acontece com a expressão “as ciências” também no mesmo exemplo.

Os fenômenos observados correspondem aos seguintes padrões:

- (i) Frases nominais definidas introduzidas por artigo demonstrativo sempre são utilizadas nos sumários como referências anafóricas. Na maioria das vezes o núcleo da expressão definida coincide com o núcleo do antecedente da frase definida como pode ser observado na Figura 3.34 onde a frase “essa metodologia” é uma referência anafórica à frase “a metodologia de ensino de Projetos de Indústrias Químicas”. Entretanto, o sintagma nominal “aquela universidade” é uma referência anafórica a “University College”; neste último caso deve ser utilizado conhecimento adicional, ou seja, o fato de “University College” ser uma universidade.

O presente artigo relata a metodologia de ensino de Projetos de Indústrias Químicas, usada no Departamento de Engenharia Química do “University College”. São mostrados os resultados obtidos **naquela universidade** (...) São apresentadas as limitações **dessa metodologia** (...) [Weyne 1987]

Figura 3.34: Referência anafórica em sumários (1)

Na Figura 3.35 a expressão definida “aquela lei” remete à expressão “a Lei 5.692/71”; notamos que o antecedente é introduzido como um nome próprio no contexto discursivo.

(...) Em épocas anteriores à reforma do ensino instituída pela Lei 5.692/71, (...) após a promulgação **daquela** lei (...) [Toropy e Aita 1987]

Figura 3.35: Referência anafórica em sumários (2)

Na Figura 3.36 a expressão anafórica “esta experiência” remete à frase “uma experiência de laboratório de graduação”; notamos que o antecedente é introduzido como uma frase indefinida que em geral se constitui no elemento inicial de uma cadeia anafórica [Mateus et al. 1983].

Neste trabalho é apresentada uma experiência de laboratório de graduação (...) **Esta experiência** é dada no (...) [Oller e Giudici 1989]

Figura 3.36: Referência anafórica em sumários (3)

Na Figura 3.37 a expressão “este tipo de programa” é uma referência anafórica à expressão “os programas de avaliação de docentes pelos alunos”. A vinculação neste caso é possível através do conhecimento lingüístico sobre a expressão “tipo de X”.

Os programas de avaliação de docentes pelos alunos vêm sendo utilizados nos Estados Unidos desde a década de quarenta. **Este tipo de programa** não é muito comum nas universidades brasileiras (...) [Milanez 1987]

Figura 3.37: Referência anafórica em sumários (4)

- (ii) Frases definidas introduzidas por artigo determinado e complementadas com participios. Em todos os casos analisados essas frases introduzem novas entidades no domínio discursivo. Isto coincide com as especificações sobre a utilização destas frases dadas por [Koch 1989, Quirk et al. 1985], segundo os quais uma frase nominal pode construir seu referente “cataforicamente” na própria frase. Assim, por exemplo, na Figura 3.38 a expressão “os resultados” não se refere ao contexto prévio e o referente é construído na própria frase.
- (iii) Frases nominais definidas cujo núcleo é derivado de um verbo e que portanto

(...) **os resultados** obtidos naquela universidade (...)
 (...) **os resultados** alcançados no Curso de Pós-Graduação
 de Indústrias Petroquímicas (...) [Weyne 1987]

Figura 3.38: Referência não anafórica em sumários (1)

não se referem ao contexto prévio devem ser consideradas como verdadeiros verbos [Pansani 1971]. Na Figura 3.39, as frases nominais cujos núcleos são, respectivamente, “utilização”, “adequação” e “formação” não se referem ao contexto prévio.

Com **a utilização** de calculadoras programáveis e micro-computadores, torna-se necessária **a adequação** do plano de ensino de Cálculo Numérico para **a formação** dos futuros engenheiros. Algumas alterações na metodologia de ensino desta disciplina são sugeridas e feitas algumas recomendações quanto a utilização de calculadoras programáveis. [de Araújo e Szeremeta 1985]

Figura 3.39: Referência não anafórica em sumários (2)

- (iv) Frases nominais com complementos preposicionados referem-se a entidades conceitualmente vinculadas a outras introduzidas previamente no texto. Assim, por exemplo, na Figura 3.39 a expressão definida “a metodologia de ensino desta disciplina” refere-se à metodologia de ensino de “Cálculo Numérico”, introduzida previamente no texto.
- (v) Frases nominais definidas que se referem ao contexto situacional. Segundo [Koch e Travaglia 1990] estas frases devem ser consideradas entidades conhecidas. No contexto de sumários, consideramos frases deste tipo: “os autores”, “o trabalho”, “o artigo”, “o objetivo do trabalho”. Na Figura 3.40 observamos a utilização deste tipo de frases definidas.

Nos textos analisados também são utilizadas referências pronominais, porém em menor frequência. Em todos os casos de referência pronominal analisados, o pronome se refere à entidade mais próxima no contexto discursivo que concorda com o pronome em gênero e número. Na Figura 3.41 o pronome “ambos” se refere a “projetores e óculos”. Na Figura 3.42 o pronome possessivo “seu” é uma referência à expressão “Universidade”. Na Figura 3.43 o pronome “a” (-la) se refere

- | |
|---|
| <p>(a) este artigo tem por finalidade mostrar (...)
[Borzani 1989]</p> <p>(b) O autor analisa as reações causa - efeito (...)
[Sperandio 1987]</p> <p>(c) Neste trabalho é apresentada uma experiência (...)
[Oller e Giudici 1989]</p> <p>(d) (...) O presente trabalho pretende descrever (...)
[Milanez 1987]</p> <p>(e) (...) O propósito deste trabalho é apresentar (...)
[Bazzo and Pereira, 1989]</p> <p>(f) (...) procura-se através deste artigo (...)
[Rodrigues 1988]</p> |
|---|

Figura 3.40: Entidades do contexto situacional

à expressão “a vida” e todos os pronomes “os” (-los) se referem a “conhecimentos”.

<p>Descreve-se um sistema de projeção tridimensional em tela metalizada utilizando-se projetores e óculos, ambos dotados de filtro polarizadores. [Eik et al. 1985]</p>
--

Figura 3.41: Referência pronominal em sumários (1)

<p>O problema do papel da Universidade é discutido (...) ressaltando-se seu apoio (...) [Forum ABENGE 1985]</p>
--

Figura 3.42: Referência pronominal em sumários (2)

Podemos sumarizar os fenômenos de referência e co-referência encontrado no corpus pelos padrões de (I) a (IV). Exemplos desses padrões são apresentado na Tabela (3.1):

- (I) Um nome próprio se refere a uma entidade do domínio discursivo; essa entidade é referenciada pela utilização de uma frase nominal definida.
- (II) Uma frase nominal não anafórica se refere a uma entidade do contexto discursivo; essa entidade é referenciada por uma frase nominal definida.

O conhecimento implica em poder, que pode contribuir para melhorar a vida como para piorá-la (...) O campo da pesquisa preocupa-se em descobrir conhecimentos, o mundo acadêmico em criticá-los e difundi-los e as empresas em aplicá-los (...)
[de Mello 1989]

Figura 3.43: Referência pronominal em sumários (3)

- (III) Uma frase indefinida introduz uma entidade no contexto; essa entidade é referenciada por uma frase nominal definida.
- (IV) Uma entidade é introduzida ou referenciada no texto; uma entidade vinculada conceitualmente a ela é referenciada.

3.9 Conclusões

Neste capítulo abordamos dois estudos que tentam tratar o problema da conexão das sentenças na seqüência textual. Por um lado vimos a coerência textual como mecanismo de vinculação conceitual entre o texto e o mundo descrito pelo texto. São várias as abordagens ao problema que mostram como as sentenças são vinculadas no texto através do descobrimento de relações entre as sentenças. Por outro lado abordamos a coesão referencial como mecanismo de conexão textual pela utilização de itens que remetem a informação introduzida no próprio texto. Descrevemos as formas básicas de coesão referencial na língua portuguesa e em particular como elas funcionam nos textos por nós analisados. A utilização de frase nominal definida anafórica mostrou-se a forma mais comum de co-referência nos sumários. Ela é particularmente usada para conectar sentenças na seqüência lingüística. A utilização de pronomes é pouco freqüente nos textos analisados e apenas restrita ao nível sentencial. Além da coesão textual como mecanismo de vinculação das sentenças no texto existe também a coesão seqüencial não abordada nesta dissertação. A partir da análise dos sumários observamos alguns padrões de utilização que servem de base para a determinação dos antecedentes das expressões anafóricas.

PADRÃO	REFERÊNCIA	CO-REFERÊNCIA
(I)	Cálculo Numérico	esta disciplina
	o University College	aquela universidade
(II)	os programas de avaliação	estes programas
	trocadores de calor	este tipo de trocador
(III)	uma experiência de laboratório	esta experiência
	um método	o método
(IV)	a disciplina	o ciclo de aulas
	a disciplina	o plano de ensino

Tabela 3.1: Padrões de referência e co-referência no corpus

Capítulo 4

Representando a estrutura de um sumário

A estrutura lingüística de um texto se refere aos princípios de organização das sentenças e das frases do texto na estrutura global [Brown e Yule 1983]. O primeiro problema abordado nessa pesquisa foi a definição de uma representação para a estrutura de um sumário em língua portuguesa. Segundo [Hobbs 1985] o discurso tem estrutura e são vários os autores que reconhecem a estrutura do discurso como composta de segmentos e relações entre segmentos. [Grosz e Sidner 1986] tratam a estrutura do discurso como um conjunto de segmentos que preenchem um determinado propósito discursivo. As relações entre os segmentos refletem relações entre os propósitos de cada segmento. [Grimes 1975, Morris e Hirst 1991] reconhecem que cada parte de um texto tem como função a transmissão de um tipo diferente de informação. [Jordan 1991, Hutchins 1977] tratam a estrutura de textos científicos como segmentos que preenchem respectivamente informações de dois tipos: (i) problema; e (ii) solução. [Pimenta Rodrigues e Pereira Lopes 1992] estudam a estrutura dos segmentos temporais de um texto e as suas relações temporais, interessados em reconhecer a coerência utilizando o conceito de tempo.

A partir da análise dos sumários observamos que o texto pode ser dividido em segmentos utilizando as informações da Figura 4.1. Um sumário poderá apenas conter algumas dessas informações. A divisão do texto produz um conjunto de segmentos, cada um contribuindo com um tipo específico de informação. O reconhecimento de cada segmento deve ser feito com base em informações de natureza lingüística, informacional e também nos padrões de produção textual observados.

Cada segmento de texto consiste de um grupo de sentenças adjacentes na superfície lingüística; as sentenças do sumário são transformadas em entidades e proposições e estas últimas são vinculadas umas às outras por relações proposicionais. Então a estrutura proposta tem três partes:

- (i) o conteúdo informacional do sumário formado pelos segmentos;

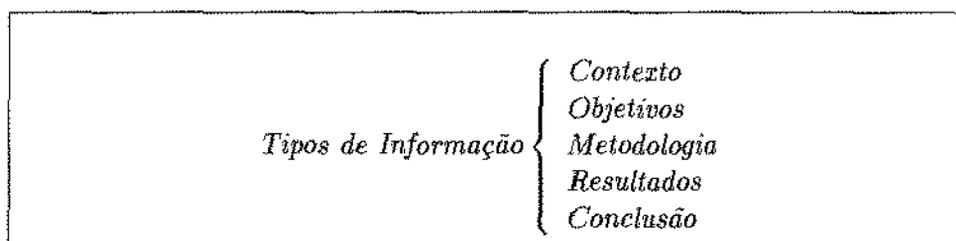


Figura 4.1: Tipo de informação em sumários

- (ii) o conteúdo proposicional do sumário formado pelas proposições e as entidades; e
- (iii) as relações proposicionais entre as proposições.

Representamos graficamente a estrutura do texto como uma árvore com nós de diferentes tipos, como pode ser visto de forma esquemática na Figura 4.2.

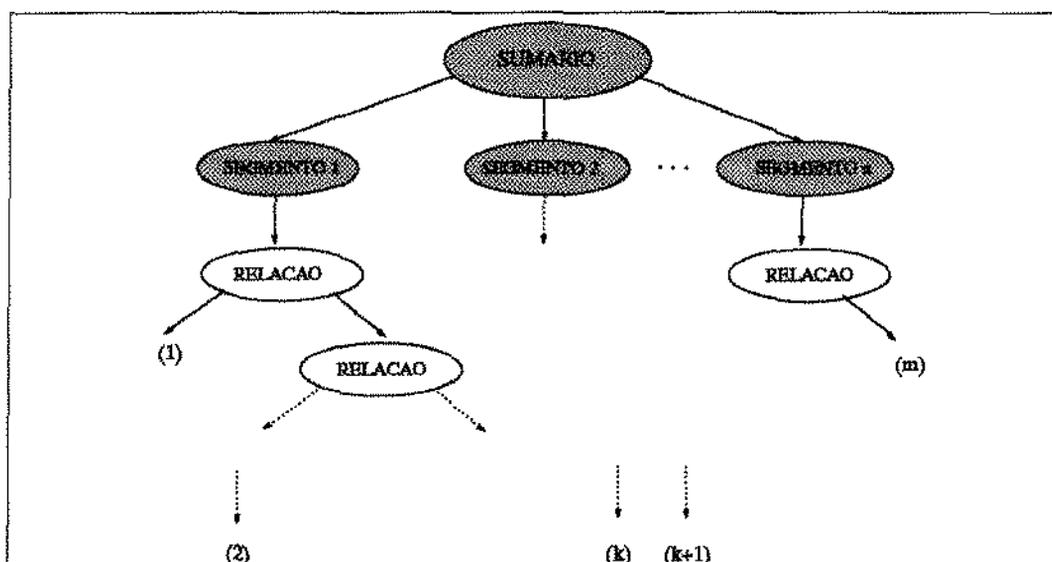


Figura 4.2: Representação esquemática da estrutura textual

A raiz da árvore representa o sumário completo e é etiquetada com o identificador “sumário”. Os filhos da raiz representam os segmentos do texto e cada nó é etiquetado com um identificador de modo a refletir o tipo de informação contido no segmento. As folhas da árvore representam o conteúdo proposicional do sumário composto pelas proposições obtidas a partir da análise das sentenças, sendo que cada proposição é etiquetada com um identificador da proposição. Os nós internos da representação indicam relações entre proposições ou grupos de proposições e cada nó é etiquetado com o nome de uma relação proposicional. Mais formalmente, um sumário é uma seqüência de segmentos informacionais que pode ser especificado por:

Sumário = Seqüência de Segmentos Informativos.

Cada segmento informativo contém duas partes: (i) o nome de uma Categoria Informativa refletindo o tipo de informação contido no segmento; e (ii) o segmento de texto do sumário. Então formalmente ele pode ser especificado por:

$$\text{Segmento Informativo} = (\text{Categoria Informativa}, \\ \text{Segmento Textual}).$$

Um segmento textual é composto de sentenças vinculadas umas as outras por relações entre sentenças; formalmente então podemos especificá-lo por:

$$\text{Segmento Textual} \left\{ \begin{array}{l} \text{Sentença.} \\ \text{Relação}(\text{Segmento}_1, \dots, \text{Segmento}_n). \end{array} \right.$$

onde $\text{Segmento}_k, 1 \leq k \leq n$ é um segmento textual. Cada sentença é considerada uma unidade composta de uma proposição ou várias proposições vinculadas por relações. Formalmente podemos especificá-la por:

$$\text{Sentença} = \text{Unidade.}$$

Finalmente, cada unidade é uma proposição ou uma relação entre proposições. Uma unidade é especificada por:

$$\text{Unidade} \left\{ \begin{array}{l} \text{Proposição} \\ \text{Relação}(\text{Unidade}_1, \dots, \text{Unidade}_n) \end{array} \right.$$

onde $\text{Unidade}_k, 1 \leq k \leq n$ é uma unidade.

Consideremos o texto apresentado na Figura 4.3.

Este sumário contém três sentenças que no texto aparecem identificadas como (S1), (S2) e (S3). Podemos observar que as sentenças (S1) e (S2) apresentam o assunto do relatório técnico; isto pode ser verificado pela utilização da expressão “neste trabalho é apresentada” na sentença (S1). Já a sentença (S2) forma com a sentença (S1) um segmento de texto. As duas sentenças estão vinculadas pelo fato delas falarem da mesma entidade. Já na sentença (S3) são introduzidos os aspectos metodológicos do trabalho desenvolvido; isto é verificado pela introdução das expressões “tratamento dos dados” e “métodos de otimização”. Concluímos então que este texto pode ser representado por dois segmentos: o primeiro contém informação que se refere aos objetivos do trabalho e o segundo contém informação que se refere à metodologia empregada no trabalho. Cada sentença por sua vez pode ser analisada detalhadamente para obter proposições, entidades e relações entre as proposições. Na Figura 4.4 podemos observar o esquema de representação deste sumário.

(S1) Neste trabalho é apresentada uma experiência de laboratório de graduação para apresentar aos alunos as técnicas experimentais de determinação de distribuição de tempos de residência em reatores químicos. (S2) Esta experiência é dada no 8^o semestre do Curso de Engenharia Química da Escola Politécnica da USP. (S3) O reator experimental é de leito fixo e o tratamento dos dados é feito por métodos de otimização usando o modelo de dispersão axial (regressão não-linear no domínio do tempo). [Oller e Giudici 1989]

Figura 4.3: Analisando um sumário

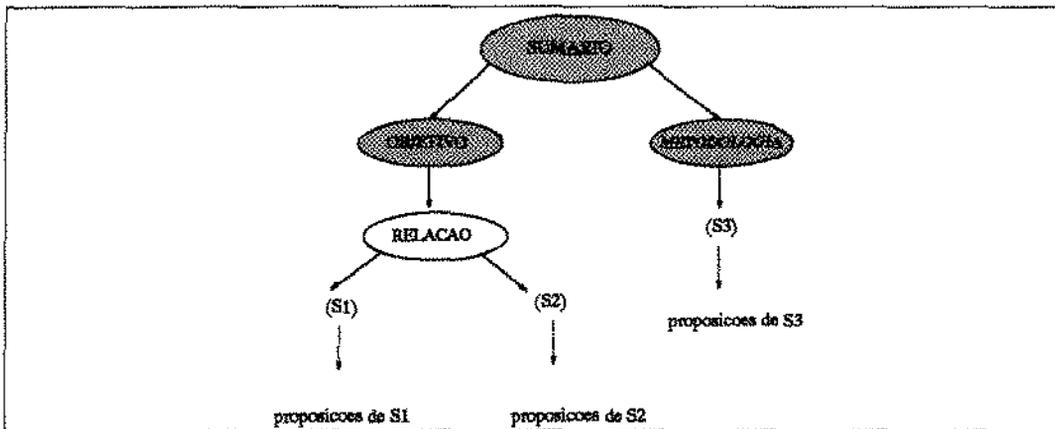


Figura 4.4: Representação parcial de um sumário

4.1 Componentes da representação

Os sumários são formados por seqüências de sentenças e estas por sua vez por frases e palavras. Este é o material lingüístico que utilizamos na especificação da representação proposta. A capacidade de um falante para criar novas sentenças opera segundo as regras de uma gramática que impõe restrições às possíveis formas superficiais. As frases de uma língua são formadas superficialmente por uma seqüência linear ordenada de itens lexicais que podemos identificar com as palavras. Os itens lexicais podem ser classificados em um número finito de *Categorias Lexicais*. No português, as Categorias Lexicais mais importantes são: *Nome (N)*, *Adjetivo (A)*, *Verbo (V)*, *Preposição (P)* e *Advérbio (Adv)*. Cada um desses elementos por sua vez é o constituinte central de uma categoria hierarquicamente superior na estrutura frasal: *Sintagma Nominal (SN)*, *Sintagma Adjetivo (SA)*, *Sintagma Verbal (SV)*, *Sintagma Preposicionado (SP)*, e *Sintagma Adverbial (SAdv)*. Uma sentença é representada pelo símbolo *S*. Existem algumas categorias menores como os *Determinantes (Det)* onde incluem-se os artigos definidos e as formas adjetivas demonstrativas; os *Quantificadores (Q)*; os *Possesivos (Pos)*; os *Pronomes Relativos (Rel)*; e as *Conjunções (Conj)*. Existem itens que pela sua função são reconhecidos como pertencentes a uma categoria sintagmática, como é o caso dos pronomes que são considerados parte da catego-

ria sintagmática *SN*.

Dá análise dos sumários detectamos alguns padrões de produção de sentenças que classificamos segundo [de Souza e Silva e Koch 1987] em:

- (i) Período simples: sentenças com um único verbo;

Exemplo 1 *Considere a sentença abaixo:*

Esta experiência é dada no 8^o semestre do Curso de Engenharia Química da Escola Politécnica da USP.

onde “é dada” é o único verbo.

- (ii) Período composto: sentenças com mais de um verbo:

- (a) Coordenação adversativa: duas sentenças independentes conectadas por uma conjunção adversativa.

Exemplo 2 *Considere a sentença:*

A engenharia depende da ciência mas não prescinde da criatividade.

onde “depende” e “prescinde” são os dois verbos da sentença e “mas” é a conjunção adversativa.

- (b) Coordenação aditiva: sentenças vinculadas pela conjunção aditiva “e”.

Exemplo 3 *Na sentença abaixo:*

O reator experimental é de leito fixo e o tratamento dos dados é feito por métodos de otimização.

onde “é” e “é feito” são os dois verbos da sentença e “e” é a conjunção aditiva.

- (c) Subordinada completiva: uma oração subordinada que é realizada na posição de um complemento do verbo.

Exemplo 4 *Considere a sentença abaixo:*

O trabalho visa apresentar a disciplina Cálculo Numérico.

onde “visa” é o verbo principal e a oração reduzida de infinitivo com verbo “apresentar” ocupa a posição de um complemento do verbo principal.

- (d) Subordinada circunstancial: uma oração subordinada anteposta à principal e realizada como uma oração reduzida de gerúndio.

Exemplo 5 *Considere a sentença:*

usando o método, os alunos podem estimar a queda da pressão.

onde a oração reduzida de gerúndio com verbo “usando” é subordinada da sentença com verbo “podem”. Esta última por sua vez contém uma oração reduzida de infinitivo com verbo “estimar”.

- (e) Subordinada relativa: uma oração introduzida pelo pronome relativo “que” e que modifica um nome.

Exemplo 6 *Considere a sentença:*

São apresentados os méritos que são a criatividade e as equipes de trabalho.

onde “são apresentados” é o verbo principal da sentença e a oração relativa introduzida pelo pronome relativo “que” é o complemento do sintagma nominal “os méritos”.

Os padrões de produção de sintagmas nominais respondem às seguintes especificações:

- (i) sintagmas nominais simples: compostos por apenas o núcleo nominal.

Exemplo 7 *Considere os nominais “dados” e “protótipos” nas frases:*

manipular dados

analisar protótipos

- (ii) sintagmas nominais com operações de determinação.

Exemplo 8 *Considere os nominais “o trabalho” e “um método” nas frases:*

o trabalho apresenta uma experiência de laboratório

o trabalho apresenta um método

- (iii) sintagmas nominais com modificadores adjetivos.

Exemplo 9 *Considere o sintagma nominal “o presente trabalho” na frase:*

o presente trabalho tem como meta divulgar a disciplina Similitude

- (iv) sintagmas nominais com operações de complementação.

Exemplo 10 *Considere as expressões “o curso de engenharia” e “a Escola Politécnica da USP” onde os núcleos nominais são complementados por sintagmas preposicionados.*

- (v) sintagmas nominais modificados por orações reduzidas de particípio.

Exemplo 11 *Considere o sintagma nominal “a disciplina ministrada no curso de engenharia” onde disciplina é modificada pela oração com particípio “ministrada”. Também o sintagma “o método recomendado neste trabalho” onde o núcleo é modificado pela oração com particípio “recomendado”.*

- (vi) sintagmas nominais com valor verbal.

Exemplo 12 *Considere o sintagma nominal “a utilização de calculadoras programáveis” onde o núcleo “utilização” (derivado do verbo utilizar) é modificado por um sintagma preposicionado considerado um complemento desse núcleo.*

As Tabelas 4.1 4.2 4.3 4.4 resumem estes casos, com seus respectivos padrões sintáticos.

<i>Sintagmas Nominais</i>	
EXEMPLO	PADRAO
dados	<i>N</i>
o artigo	<i>Det N</i>
o presente trabalho	<i>Det A N</i>
a disciplina Similitude em Engenharia	<i>Det N N</i>
o curso de engenharia	<i>Det N SP</i>
fundamentos básicos da teoria de modelos	<i>N A SP</i>
a disciplina ministrada no curso de engenharia	<i>Det N SA</i>

Tabela 4.1: Padrões nominais encontrados no corpus de sumários

4.2 Representação das entidades

As entidades a serem representadas são obtidas a partir dos nominais, já que são essas expressões da língua portuguesa as caracterizáveis semanticamente por terem potencial de referência. Cada sintagma nominal é mapeado para uma expressão que construímos utilizando os seguintes conjuntos de símbolos:

<i>Sintagmas Verbais</i>	
EXEMPLO	PADRÃO
depende das ciências	$V SP$
visa proporcionar aos alunos fundamentos	$V S_{inf}$
proporcionar aos alunos fundamentos	$V_{inf} SP SN$

Tabela 4.2: Padrões verbais encontrados no corpus de sumários

<i>Sentença Simples (Período Simples)</i>	
EXEMPLO	PADRÃO
a engenharia depende da ciência	$SN SV$
trocadores de calor são elementos básicos	$SN SV$

Tabela 4.3: Padrões sentenciais simples encontrados no corpus de sumários

<i>Sentença Composta (Período Composto)</i>	
EXEMPLO	PADRÃO
usando o método, os alunos podem estimar a queda da pressão	$S_{ger} S$
alterações são sugeridas e são feitas recomendações	$S Conj S$

Tabela 4.4: Padrões sentenciais compostos encontrados no corpus de sumários

- (i) $P_{nominais}$: os predicados nominais;
- (ii) $C_{nominais}$: as constantes nominais;
- (iii) $C_{próprios}$: as constantes de nomes próprios;
- (iv) $C_{adjetivos}$: as constantes adjetivas;
- (v) $C_{determinação}$: as constantes de determinação;
- (vi) $C_{gênero} = \{mas, fem, none\}$; as constantes de gênero;
- (vii) $C_{número} = \{sin, plu\}$; as constantes de número; e
- (viii) $I_{entidades}$: conjunto de índices para identificar as entidades na estrutura do sumário.

Como já foi visto um sintagma nominal no português é composto de: (i) um núcleo nominal; (ii) adjetivos que modificam o núcleo nominal; (iii) nomes próprios atribuídos ao núcleo nominal; e (iv) complementos nominais introduzidos por preposição. Cada um desses componentes deve estar presente na representação da entidade.

Um sintagma nominal com m modificadores será representado por uma expressão do tipo (4.1)

$$p(i, e_1, \dots, e_m, d, g, n) \quad (4.1)$$

onde:

- (a) $p \in P_{nominais}$ é o predicado nominal associado ao núcleo nominal;
- (b) $i \in I_{entidades}$ é um índice que identifica a entidade ou conceito descrito pelo sintagma nominal. Esse índice será utilizado na estrutura argumental dos verbos;
- (c) e_k é uma expressão que representa o k -ésimo complemento do núcleo nominal;
- (d) $d \in C_{determinação}$ é uma expressão associada com a operação de determinação aplicada ao núcleo nominal;
- (e) $g \in C_{gênero}$ é uma expressão associada com a informação sobre gênero; e
- (f) $n \in C_{número}$ é uma expressão associada com a informação sobre número.

A expressão d depende das operações de determinação e quantificação aplicadas ao núcleo nominal. Adotamos o seguinte mapeamento para os determinantes:

- (i) Os artigos determinados $\{o, a, os, as\}$ são representados pela constante det ;

- (ii) Os artigos não determinados $\{um(a), uns, umas\}$ são representados pela constante *nodet*;
- (iii) Os artigos demonstrativos $\{este(s), esta(s)\}$ são representados pela constante *dem1*;
- (iv) Os artigos demonstrativos $\{esse(s), essa(s)\}$ são representados pela constante *dem2*;
- (v) Os artigos demonstrativos $\{aquele(s), aquela(s)\}$ são representados pela constante *dem3*;
- (vi) Os quantificadores $\{algum(s), alguma(s)\}$ são representados pela constante *algum*; e
- (vii) Os quantificadores $\{todo(s), toda(s)\}$ são representados pela constante *todo*;

A expressão associada ao *i*-ésimo complemento nominal é construída a partir do tipo de elemento sintático, como veremos a seguir.

Adjetivos

Um adjetivo *A* será representado pela expressão *qualificador* = *A'*, onde *A'* ∈ *C_{adjetivo}* é a representação do adjetivo *A* na linguagem de representação.

Exemplo 13 O adjetivo “*simples*” na representação do sintagma nominal “*um método simples*” será representado pela expressão:

$$\textit{qualificador} = \textit{SIMPLES}.$$

Os nominais simples introduzidos por preposição

Um nome *N* sem complementação que modifica o núcleo nominal será representado pela expressão *modificador* = *N'*, onde *N'* ∈ *C_{nominais}* é a representação do nome *N* na linguagem de representação. Apenas representamos modificadores nominais introduzidos pela preposição “*de*”.

Exemplo 14 O nome “*laboratório*” na representação do sintagma nominal “*o trabalho de laboratório*” será representado pela expressão:

$$\textit{modificador} = \textit{LABORATORIO}.$$

Nomes próprios

Um nome próprio N será representado por uma expressão do tipo $n(i, nome = N')$, onde $N' \in C_{próprios}$, $n \in P_{nominais}$ e $i \in I_{nominais}$.

Exemplo 15 O nome próprio “UFU” no sintagma nominal “o curso da UFU” será representado pela expressão:

$$universidade(i_k, nome = ufu, det, fem, sin).$$

Nominais com operação de determinação

Para um sintagma nominal composto que modifica um núcleo será construída uma expressão como (4.1). O índice $j \in I_{nominais}$ utilizado na construção dessa representação aparecerá como argumento na representação do núcleo do sintagma nominal. Será utilizado o termo *especificador* = j na representação completa.

Exemplo 16 Na expressão “coeficientes de troca de calor”, o sintagma nominal “troca de calor”, será representado pela expressão

$$especificador = i_l$$

e será construída a expressão:

$$troca(i_l, modificador = CALOR, none, fem, sin).$$

Exemplo 17 O sintagma nominal “o curso de engenharia da UFU” será representado pela expressão

$$curso(i_p, modificador = ENGENHARIA, especificador = i_q, det, mas, sin)$$

e também será construída a expressão:

$$universidade(i_q, nome = ufu, det, fem, sin).$$

Exemplo 18 O sintagma nominal “a disciplina Similitude em Engenharia” será representado pela expressão:

$$disciplina(i_d, nome = similitude_em_engenharia, det, fem, sin).$$

4.2.1 Representação das anáforas

Os pronomes de terceira pessoa são representados pela expressão (4.2)

$$\text{anafora}(i, g, n) \quad (4.2)$$

onde:

- (i) $i \in I_{\text{nominais}}$ identifica o referente da anáfora;
- (ii) $g \in C_{\text{gênero}}$; e
- (iii) $n \in C_{\text{número}}$.

Exemplo 19 O pronome “ele” é representado pela expressão

$$\text{anafora}(i_1, \text{mas}, \text{sin})$$

onde i_1 é o índice associado ao antecedente do pronome no texto.

4.3 Representação das proposições

No português as relações entre entidades são expressas mediante elementos predicativos que descrevem propriedades sobre as entidades ou relacionam duas ou mais entidades.

Consideramos elementos predicativos qualquer uma das seguintes formas de um verbo: (i) infinitivo; (ii) indicativo; (iii) particípio; (iv) gerúndio; (v) alguns adjetivos com valor predicativo; e (vi) os nominais derivados de um verbo.

Cada elemento predicativo produz uma proposição. Dado que uma sentença do português pode ser composta de mais de um elemento predicativo (orações subordinadas e coordenadas), uma sentença poderá produzir mais de uma proposição.

A cada predicativo é associado um conjunto de argumentos e existem predicadores de um, dois ou três lugares. Os argumentos exigidos pelo elemento predicativo são chamados nucleares. Alguns elementos realizados na sentença e que complementam o significado da frase são considerados opcionais. Os argumentos obrigatórios devem estar presentes explicitamente, ou serem recuperáveis do contexto.

Para representar cada proposição adotamos os seguintes conjuntos de símbolos:

- (i) P_{verbais} : os predicados verbais;

- (ii) $C_{argumentais} = \{ Paciente, Neutro, Origem, Objeto, Experienciador, Recipiente, Locativo, Agente, Posicionador, Instrumento, Beneficiário, Causa, Fim \}$: constantes que indicam a função semântica do argumento com o predicador;
- (iii) $C_{tempo} = \{ pas, pre, fut \}$: constantes que indicam o tempo verbal do predicador;
- (iv) $C_{modo} = \{ ind, inf, ger, par \}$: constantes que indicam a modalidade do predicador;
- (v) $C_{aspecto} = \{ acabado, inacabado \}$: constantes que indicam a informação aspectual do predicador; e
- (vi) $I_{verbais}$: um conjunto de índices para identificar a proposição na estrutura do sumário.

Cada proposição com n argumentos na superfície lingüística será representada mediante uma expressão do tipo (4.3)

$$p(i, papel_1 = e_1, \dots, papel_n = e_n, t, m, a) \quad (4.3)$$

onde:

1. $p \in P_{verbais}$ é o predicador associado ao elemento predicativo;
2. $i \in I_{verbais}$ é o índice que identifica a proposição na estrutura do sumário;
3. $papel_j \in C_{argumentais}$ é a função semântica do j -ésimo argumento do predicador;
4. e_j é o índice do j -ésimo elemento relacionado pelo predicador;
5. $t \in C_{tempo}$: indica o tempo verbal do predicador;
6. $m \in C_{modo}$: indica o modo do predicador; e
7. $a \in C_{aspecto}$: indica o aspecto do predicador.

Exemplo 20 A sentença “a engenharia depende das ciências” será representada pela expressão:

$$depende(p_1, paciente = i_1, neutro = i_2, pre, ind, acabado).$$

As entidades “a engenharia” e “as ciências” serão representadas por:

$$\begin{aligned} &engenharia(i_1, det, fem, sin) \\ &ciencia(i_2, det, fem, plu). \end{aligned}$$

4.4 Representação de nominais derivados de um verbo

Os nominais derivados de um verbo são representados de uma forma especial. Esses nominais possuem um conjunto de posições predefinido que deve aparecer na representação. Um nominal derivado de um verbo será representado por uma expressão como (4.4)

$$p(i, papel_1 = e_1, \dots, papel_m = e_m, d, g, n) \quad (4.4)$$

onde:

1. $p \in P_{verbais}$ é o predicador associado ao núcleo nominal;
2. $i \in I_{verbais}$ é o índice que identifica a proposição;
3. $papel_k \in C_{argumentais}$ é a função semântica do k -ésimo complemento do nome;
4. e_k é o índice do k -ésimo complemento do nome;
5. $d \in C_{determinação}$;
6. $g \in C_{gêneros}$ e
7. $n \in C_{número}$.

Exemplo 21 A expressão nominal “a utilização de calculadoras programáveis” será representada pela expressão

$$utilizar(e_k, objeto = i_j, det, fem, sin)$$

onde i_j é o índice utilizado na representação do nominal “calculadoras programáveis”:

$$calculadora(i_j, modificador = PROGRAMAVEL, none, fem, plu).$$

4.5 Representação dos advérbios

Representamos os advérbios que modificam um predicativo por uma expressão do tipo (4.5). Utilizamos um conjunto $P_{advérbios}$ que contém os predicadores associados aos advérbios e um conjunto $I_{advérbios}$ para identificar a expressão na representação

$$p(i, j) \quad (4.5)$$

onde:

- (i) $p \in P_{advérbios}$;

(ii) $i \in I_{adverbios}$; e

(iii) $j \in I_{verbais}$

Exemplo 22 O advérbio de negação “não” que modifica o verbo na sentença “este tipo de programa não é comum nas universidades brasileiras”, será representado por

$$nao(i_k, j_n)$$

onde j_n é o índice associado à proposição produzida a partir da sentença que está sendo negada.

4.6 Representação das relações proposicionais

No texto as proposições são expressas por meio de sentenças e uma sentença pode vincular sintaticamente mais de uma proposição, utilizando construções subordinadas e coordenadas.

Estamos interessados na vinculação das proposições utilizando um conjunto de *Relações Proposicionais* divididas em dois grupos:

- (i) Relações determinadas a partir dos vínculos coesivos entre entidades do texto, como por exemplo as relações *Elaboração* e *Paralelo*; e
- (ii) Relações sintaticamente marcadas que representam relações lógico-semânticas, como por exemplo as relações *Seqüência*, *CausaConseqüência*, *Contraste*, *SeqüênciaTemporal*, e *Ocasião*.

As relações proposicionais por nós identificadas nos sumários foram:

$$RP = \{ \textit{Elaboração}, \textit{Paralelo}, \textit{Seqüência}, \textit{CausaConseqüência}, \textit{Contraste}, \textit{SeqüênciaTemporal}, \textit{Ocasião} \}$$

Cada relação que vincula k elementos será representada por uma expressão do tipo (4.6)

$$r(i, l_1, \dots, l_k) \quad (4.6)$$

onde $r \in RP$, l_j é o índice do elemento (relação ou proposição) vinculado pela relação r ; $i \in I_{RP}$ é o índice que identifica à relação na representação do sumário.

Relação de Elaboração

A relação de *Elaboração* vincula duas proposições p_i e p_j se a proposição p_i introduz uma entidade e_k e a proposição p_j elabora a entidade e_k ou uma entidade e_l relacionada conceitualmente com e_k .

Exemplo 23 *Consideremos a seguinte seqüência de sentenças:*

O presente trabalho tem como meta divulgar a disciplina Similitude em Engenharia. Esta disciplina visa proporcionar aos alunos fundamentos básicos sobre a teoria de modelos.

Neste exemplo a primeira sentença introduz a entidade “a disciplina Similitude em Engenharia” pela primeira vez no texto. A segunda sentença faz uma elaboração da mesma entidade, realizada linguisticamente com uma referência definida “esta disciplina”. Por isso, as duas são vinculadas por uma relação de Elaboração.

Relação de Paralelo

A relação de *Paralelo* é utilizada na vinculação de duas ou mais proposições que partilham elementos conceitualmente relacionados a uma entidade introduzida previamente no texto.

Exemplo 24 *Consideremos esta sentença de um sumário do corpus:*

Este trabalho apresenta um método simples usado para o desenvolvimento, teste e análise térmica deste tipo de trocador.

Podemos observar que existe uma construção coordenada realizada por vírgula “,” e pela conjunção “e”; os elementos da coordenação partilham o elemento “este tipo de trocador” que foi introduzido previamente no texto. Por isso essa construção é classificada como Paralelo.

Relação de Seqüência

A relação de *Seqüência* vincula duas ou mais proposições considerando apenas construções sintáticas coordenadas ou justaposição na realização lingüística da seqüência.

Exemplo 25 *Consideremos a construção coordenada extraída do corpus:*

O ciclo de aulas práticas tem como objetivos distinguir as quantidades envolvidas em um experimento, manipular dados e analisar protótipos através de modelos.

A coordenação utilizada no texto e a ausência de outro tipo de informação possibilita o reconhecimento de uma seqüência nas proposições: (i) distinguir as quantidades envolvidas em um experimento; (ii) manipular dados; e (iii) analisar protótipos através de modelos. Por isso, as proposições são vinculadas por uma relação de Seqüência.

Relação de Seqüência Temporal

A relação de *Seqüência Temporal* foi determinada no corpus dos sumários observando a utilização de construções subordinadas com advérbios de tempo.

Exemplo 26 *Consideremos o fragmento de sumário abaixo:*

Os autores, após fazerem uma análise, partem para um estudo visando encontrar soluções.

Neste exemplo o advérbio temporal após marca uma vinculação entre as proposições: (i) fazer uma análise; e (ii) partir para um estudo visando encontrar soluções. Essa vinculação é representada por uma relação de Seqüência Temporal.

Relação de Contraste

A relação de *Contraste* é reconhecida pela utilização de conjunções coordenadas adversativas em uma construção sintática de coordenação.

Exemplo 27 *Consideremos a construção coordenada abaixo:*

A engenharia depende das ciências, mas jamais vai prescindir do empirismo e da criatividade.

Neste exemplo a construção coordenada utilizando mas indica uma relação de contraste entre as duas proposições. Por isso, as proposições são vinculadas por uma relação de Contraste.

Relação de Causa Conseqüência

A relação de *Causa Conseqüência* vincula duas proposições que lingüisticamente estão vinculadas por uma construção sintática de causa-conseqüência.

Exemplo 28 *Observemos a sentença complexa tomada do corpus:*

Com a utilização de calculadoras programáveis, torna-se necessária a adequação do plano de ensino de Cálculo Numérico.

A utilização de uma construção subordinada introduzida pela preposição com marca uma proposição causal quando utilizada com um verbo como "tornar". Portanto as proposições serão vinculadas utilizando a relação de Causa Conseqüência.

Relação de Ocasão

A relação de *Ocasão* vincula duas proposições também baseando-se em construções sintáticas subordinadas circunstanciais.

Exemplo 29 *Consideremos a sentença do corpus:*

Usando o método recomendado, os alunos podem estimar a queda da pressão.

No exemplo a oração reduzida de gerúndio “usando o método recomendado” marca este tipo de relação; portanto as proposições serão vinculadas pela relação de Ocasão.

4.7 Representação dos segmentos

Baseando-nos nas informações teóricas da ciência da documentação sobre o tipo de material contido em um sumário e a partir da análise dos textos identificamos o seguinte conjunto de *Categorias Informacionais (CI)* para a classificação de cada segmento. O nome da categoria reflete o tipo de material contido no segmento e cada sumário poderá apenas conter um subconjunto próprio do conjunto de *CI*. As categorias são as seguintes:

$$CI = \{ \textit{Contexto}, \textit{Objetivo}, \textit{Método}, \textit{Experimento}, \textit{Resultado}, \textit{Conclusão} \}.$$

Cada segmento é representado utilizando-se um termo do tipo (4.7)

$$\textit{categoria}(i) \tag{4.7}$$

onde *Categoria* \in *CI* e *i* é o índice que identifica o segmento de texto pertencente à categoria *Categoria* (índice de uma proposição ou de uma relação proposicional). Na Figura 4.5 são apresentados exemplos de cada tipo de segmento encontrado nos textos analisados.

4.8 Exemplos de proposições e relações

Consideremos o segmento de sumário apresentado na Figura 4.6. Para representarmos essa seqüência textual na estrutura proposta devemos considerar por um lado os nominais, e por outro lado os elementos predicativos.

Os nominais da seqüência são os seguintes:

- (i) “o artigo” que identificamos com o índice $x_1 \in I_{\textit{nominais}}$;
- (ii) “Projetos de Indústrias Químicas” que identificamos com o índice $x_3 \in I_{\textit{nominais}}$;

- (i) Contexto
*Constata-se hoje uma **deficiência** no ensino de desenho na escola brasileira.*
- (ii) Objetivo
***O presente artigo relata** a metodologia de ensino de Projetos de Indústrias Químicas.*
- (iii) Metodologia
***O corpo teórico** que fundamentou o estudo focaliza a literatura pertinente à área e o modelo FTC.*
- (iv) Resultado
Os primeiros resultados da investigação evidenciam a necessidade de treinamento.
- (v) Experimento
No departamento de Engenharia Mecânica da UNICAMP foram feitas algumas tentativas de implantação destes programas.
- (vi) Conclusão
São apresentadas as limitações e os méritos dessa metodologia.

Figura 4.5: Segmentos no texto

*O artigo relata a metodologia de ensino de Projetos de Indústrias Químicas usada no Departamento de Engenharia Química do “University College”.
[de Sá Weyne 1987]*

Figura 4.6: Segmento de sumário

artigo	<i>artigo</i>
Projetos de Indústrias Químicas	<i>disciplina(x₃, nome = piq)</i>
metodologia	<i>metodologia</i>
ensino	<i>ENSINO</i>
University College	<i>universidade(x₅, nome = uc)</i>
Departamento de Engenharia Química	<i>departamento(x₄, nome = deq)</i>
relata	<i>relatar</i>
usada	<i>usar</i>

Tabela 4.5: Mapeamento dos elementos lexicais

- (iii) “a metodologia de ensino de Projetos de Indústrias Químicas” que identificamos com o índice $x_2 \in I_{nominais}$;
- (iv) “University College” que identificamos com o índice $x_5 \in I_{nominais}$;
- (v) “o Departamento de Engenharia Química do University College” que identificamos com o índice $x_4 \in I_{nominais}$.

Os elementos predicativos são:

- (i) “relata” que identificamos com o índice $y_1 \in I_{verbais}$;
- (ii) “usada” que identificamos com o índice $y_2 \in I_{verbais}$.

A análise da seqüência textual produz duas proposições e cinco entidades. Essas duas proposições estão coesivamente vinculadas no texto em uma sentença só. A relação entre as proposições é capturada na representação pela relação de *Elaboração*. O termo elaborado é justamente “a metodologia de ensino de Projetos de Indústrias Químicas”.

Para representar as cinco entidades fazemos o mapeamento entre itens lexicais e elementos da representação apresentado na Tabela 4.5.

As entidades são representadas através dos termos de (4.8) a (4.12):

$$\textit{artigo}(x_1, \textit{det}, \textit{mas}, \textit{sin}) \quad (4.8)$$

$$\begin{aligned} \textit{metodologia}(x_2, \textit{modificador} = \textit{ENSINO}, \\ \textit{especificador} = x_3, \textit{det}, \textit{fem}, \textit{sin}) \end{aligned} \quad (4.9)$$

$$\text{disciplina}(x_3, \text{nome} = \text{piq}, \text{det}, \text{fem}, \text{sin}) \quad (4.10)$$

$$\text{departamento}(x_4, \text{nome} = \text{deq}, \text{especificador} = x_5, \text{det}, \text{mas}, \text{sin}) \quad (4.11)$$

$$\text{universidade}(x_5, \text{nome} = \text{uc}, \text{det}, \text{fem}, \text{sin}) \quad (4.12)$$

e as proposições são representadas pelos termos (4.13) e (4.14)

$$\text{relatar}(y_1, \text{origem} = x_1, \text{neutro} = x_2, \text{pre}, \text{ind}, \text{acabado}) \quad (4.13)$$

$$\text{usar}(y_2, \text{paciente} = x_2, \text{locativo} = x_4, \text{none}, \text{par}, \text{none}) \quad (4.14)$$

A relação de *Elaboração* entre as proposições é representada pelo termo (4.15)

$$\text{Elaboração}(i_1, y_1, y_2) \quad (4.15)$$

onde $i_1 \in IRP$.

Supondo que a informação do segmento é de tipo objetivo, podemos representar este segmento em forma de árvore como pode ser observado na Figura 4.7, ou como o termo (4.16)

$$\begin{aligned} \text{Sumário} = & (\text{Objetivo Elaboração} \\ & (\text{relatar}(y_1, \text{origem} = x_1, \text{neutro} = x_2, \text{pre}, \text{ind}, \text{acabado})) \\ & (\text{usar}(y_2, \text{paciente} = x_2, \text{locativo} = x_4, \text{none}, \text{par}, \text{none})) \\ &)) \dots \end{aligned} \quad (4.16)$$

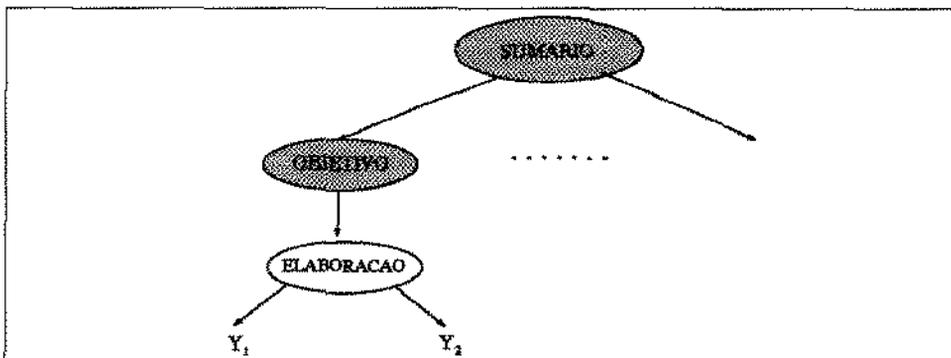


Figura 4.7: Representação de um segmento de sumário

4.9 Representações

Nesta seção apresentamos três exemplos de análise de sumários visando a construção da representação proposta.

Na Figura 4.8 podemos observar um sumário e a sua representação em uma estrutura de árvore.

Os números no sumário representam orações; essa partição é obtida a partir dos predicativos da língua portuguesa. Neste texto podemos observar que o autor inclui três tipos diferentes de informação, a saber:

- Nas orações de número (1), (2) e (3) o autor estabelece a necessidade de adequar o plano de ensino de Cálculo Numérico. Esta informação indica o problema levantado no artigo e portanto trata-se de informação de tipo contextual.
- Na oração de número (4) notamos a utilização do verbo “sugerir” na expressão “são sugeridas”. Essa oração indica que no texto do respectivo artigo sugestões serão apresentadas sobre a metodologia de ensino de Cálculo Numérico. Portanto essa oração indica informação de tipo sugestão.
- Finalmente, na oração de número (5) é utilizado o nominal “recomendações”; esse item lexical informa que no artigo serão mostradas recomendações sobre o uso de calculadoras. Portanto essa oração indica material de tipo recomendação.

(1) Com a utilização de calculadoras programáveis e microcomputadores, (2) torna-se necessária a adequação do plano de ensino de Cálculo Numérico (3) para a formação dos futuros engenheiros. (4) Algumas alterações na metodologia de ensino desta disciplina são sugeridas e (5) feitas algumas recomendações quanto à utilização de calculadoras programáveis.

Figura 4.8: Exemplo de sumário (1)

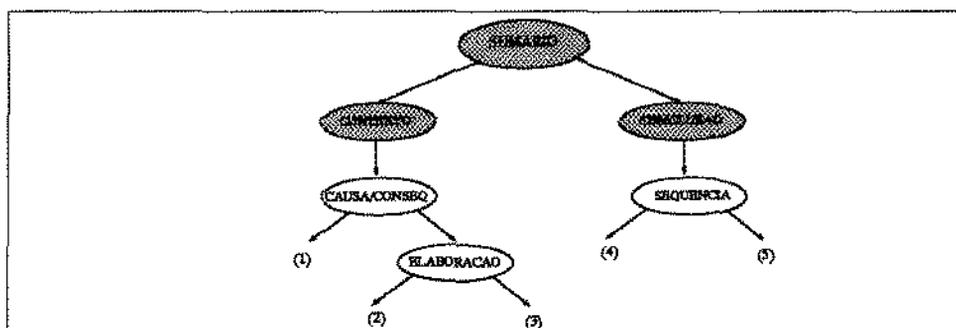


Figura 4.9: Representação do sumário (1)

Então a estrutura proposta para esse sumário consiste de dois segmentos, como pode ser observado na Figura 4.9. O primeiro segmento formado pelas

orações (1), (2) e (3); o segundo segmento formado pelas orações (4) e (5).

Como o primeiro segmento contém três orações, elas serão vinculadas utilizando relações entre proposições como pode ser visto na Figura 4.9.

Entre as orações (2) e (3) existe uma vinculação marcada sintaticamente pela preposição “para” indicando uma elaboração nos objetivos da “adequação” introduzida na oração (2). Essas duas orações formam um segmento. Entre a oração (1) e o segmento formado pelas orações (2) e (3) pode ser verificada a relação de “causa” marcada sintaticamente pela construção subordinada introduzida pela preposição “com” e o uso do predicador “necessária”.

O segundo segmento é de tipo conclusão, marcado pelas informações de tipo sugestão e recomendação. Portanto as duas orações (4) e (5) são vinculadas por uma relação de *Seqüência*. A análise apresentada possibilita a construção da estrutura do texto em forma de árvore.

Consideremos agora o sumário da Figura 4.10 que contém quatro orações.

(1) *Os programas de avaliação de docentes pelos alunos vêm sendo utilizados nos Estados Unidos desde a década de quarenta.*
 (2) *Este tipo de programa não é muito comum nas universidades brasileiras.* (3) *No Departamento de Engenharia Mecânica da UNICAMP foram feitas algumas tentativas de implantação destes programas.* (4) *O presente trabalho descreve a experiência da aplicação destes programas no Departamento de Engenharia Mecânica da UNICAMP.*

Figura 4.10: Exemplo de sumário (2)

Este texto também pode ser dividido em três partes, cada uma refletindo um tipo de informação:

- As orações (1) e (2) compõem o primeiro segmento de texto que apresenta informação de tipo contextual. As orações introduzem informação sobre a entidade da qual trata o artigo, ou seja “os programas de avaliação”.
- O segundo segmento do texto é formado pela oração (3); este segmento contém informação de tipo experimento, já que se refere a uma experiência de implantação. Podemos notar o uso do tempo passado no segmento.
- O terceiro segmento é formado pela oração (4) que indica o tópico principal do artigo; consideramos este tipo de informação como objetivo. Isto é verificado pela utilização do verbo “descrever” e do sintagma nominal “o presente trabalho” que na realidade significa “o presente artigo”.

A estrutura textual consiste de três segmentos, como pode ser observado na Figura 4.11. Notamos que as orações (1) e (2) estão contidas no mesmo segmento e, portanto, devem ser vinculadas na representação. A vinculação neste caso é realizada por uma relação de *Elaboração*. A elaboração é verificada pela referência

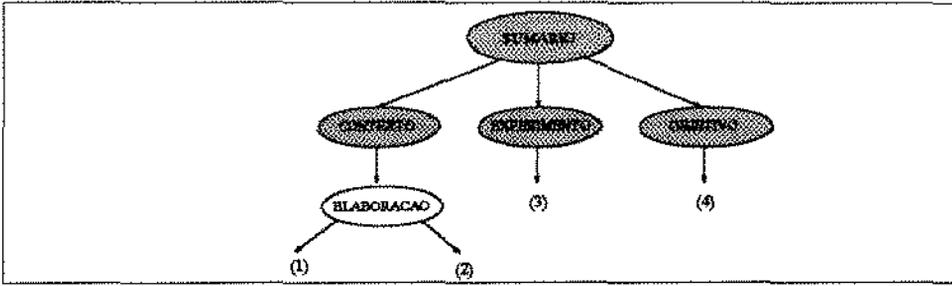


Figura 4.11: Representação do sumário (2)

a mesma entidade “os programas de avaliação” nas duas orações.

Finalmente consideremos o sumário da Figura 4.12.

(1) A engenharia depende das ciências, mas (2) jamais vai prescindir da criatividade. (3) Na UFSC, esta sendo realizada uma experiência (4) visando desenvolver o uso da criatividade na engenharia. (5) Este trabalho apresenta resultados de experiências (6) realizadas com estudantes (7) para o desenvolvimento de trabalhos na engenharia.

Figura 4.12: Exemplo de sumário (3)

Podemos determinar neste texto tres segmentos, como pode ser observado na Figura 4.13, a saber:

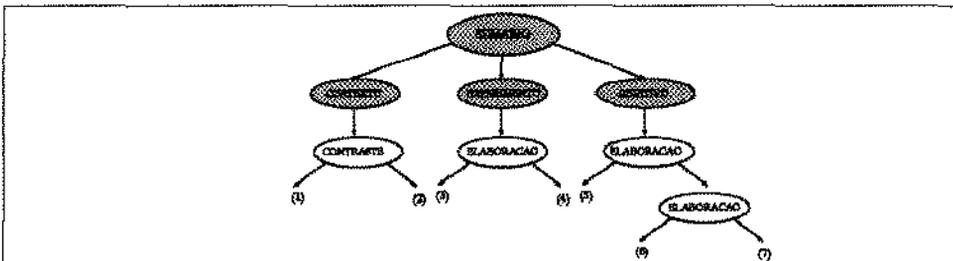


Figura 4.13: Representação do sumário (3)

- Nas orações (1) e (2) um fato é apresentado, portanto esse segmento contém informação de tipo contextual.
- Nas orações (3) e (4) informação sobre uma experiência científica é introduzida. Este segmento indica informação de tipo experimento.
- Nas orações (5), (6) e (7) é apresentado o assunto principal do artigo; portanto esse segmento é de tipo objetivo.

No primeiro segmento as orações (1) e (2) estão vinculadas por uma construção coordenada adversativa; essa construção é classificada como *Contraste*. As orações (3) e (4) estão vinculadas pela relação de *Elaboração*. Finalmente as orações (5), (6) e (7) do terceiro segmento são vinculadas por elaborações.

4.10 Conclusões

Neste capítulo abordamos a primeira questão apontada nesta dissertação, ou seja, a definição e representação da estrutura dos sumários. As sentenças do sumário são o material lingüístico básico para a especificação da representação. A representação proposta foi obtida a partir do estudo de textos reais e da verificação de relações entre sentenças. A representação consiste de três partes fundamentais: (i) **o conteúdo proposicional** do sumário refletido nas proposições e as entidades; (ii) **as relações entre proposições e sentenças** no texto, definidas por vínculos sintáticos e relações de coesão; e (iii) a determinação de **segmentos de texto**, cada um contendo um tipo de informação obtida a partir do estudo do corpus. A estrutura do texto pode ser representada em forma de árvore, onde as folhas representam o conteúdo proposicional, os nós internos representam as relações entre segmentos de texto e os nós filhos da raiz representam os segmentos textuais. A árvore toda representa a estrutura do texto. Decidimos representar as entidades e as proposições utilizando predicados e argumentos que contém informações de tipo sintático-semântico. A representação das entidades é baseada nos componentes dos sintagmas nominais, a representação das proposições é baseada nos predicativos e na sua estrutura argumental. As relações entre proposições foram especificadas a partir de construções sintáticas ou de vínculos coesivos verificados no corpus. Os segmentos foram determinados a partir dos componentes básicos de um sumário e dos padrões observados nos textos.

Capítulo 5

Construção da representação de um sumário

A análise de um texto é o processo pelo qual as sentenças são transformadas em proposições que serão vinculadas umas as outras em uma representação estrutural. Neste capítulo apresentamos os processos que tornam possível a vinculação de sentenças e proposições na representação textual definida.

A arquitetura do analisador textual proposto pode ser observada na Figura 5.1.

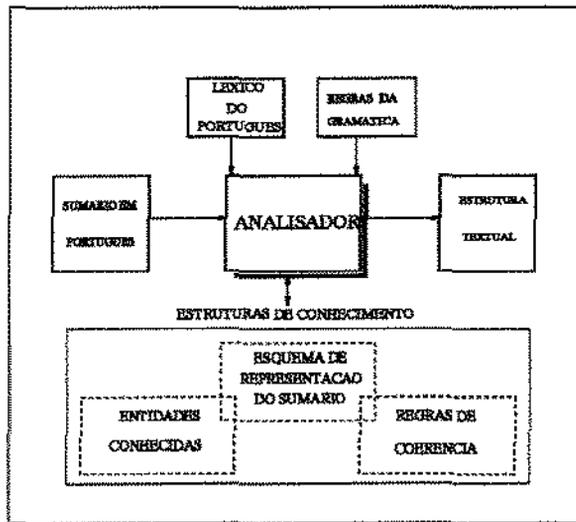


Figura 5.1: Arquitetura do analisador textual

O texto do sumário é analisado com o objetivo de construir uma representação estrutural. As fontes de conhecimento envolvidas são: (i) o léxico da língua; (ii) a gramática da língua; (iii) as entidades conhecidas para a resolução das referências; (iv) as regras de vinculação entre entidades do domínio; e (v) dados sobre a estrutura de informação do sumário. A representação textual proposta nesta dissertação, que é a saída do analisador, contém três componentes fundamentais

que podemos observar na Figura 5.2.

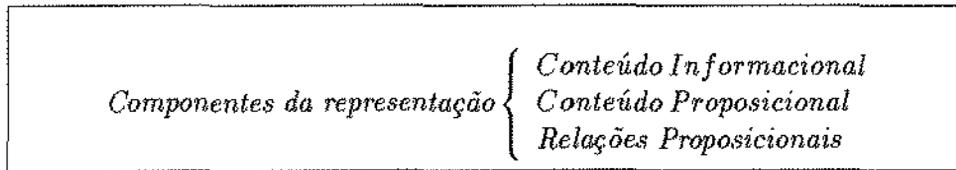


Figura 5.2: Componentes da representação

Para cada componente dessa representação, um processo computacional deve ser especificado. Um analisador a nível sentencial é o encarregado de produzir um conjunto de entidades e proposições a partir das sentenças. Para isso utiliza-se do léxico da língua e das regras da gramática. Um processo de interpretação das referências é incorporado ao analisador para a resolução dos pronomes e das frases nominais definidas. Estes processos possibilitam o cálculo do **Conteúdo Proposicional** do texto. Todas as proposições produzidas a partir da análise de uma sentença serão vinculadas na representação. Essa vinculação pode ser realizada na fase de análise da sentença, utilizando informação sobre a estrutura argumental de um verbo, ou utilizando as próprias construções sintáticas que marcam relações lógico-semânticas. Então na fase de análise das sentenças podem ser calculadas também algumas das **Relações Proposicionais** entre proposições. Cada sentença analisada deve ser a seguir classificada visando determinar o segmento ao qual ela pertence. A classificação é baseada nos predicadores utilizados na sentença. Este processo possibilita a determinação dos segmentos do sumário que representam o **Conteúdo Informacional** do texto. Todas as proposições em um segmento devem ser vinculadas utilizando-se informação de tipo coesiva obtida a partir do processo de interpretação das referências. Neste passo é determinado o restante das Relações Proposicionais no segmento. O processamento é baseado nos padrões de produção textual observados nos sumários analisados. Nas próximas seções cada um desses processos será detalhadamente descrito.

5.1 Construindo entidades e proposições

Os nominais no português são as expressões caracterizáveis semanticamente por terem potencial de referência [Raposo 1992]; são essas as expressões que descrevem as entidades que serão representadas no formalismo de representação. As outras expressões lingüísticas que no português têm poder de referência são as orações que denotam relações entre entidades. Esse é o segundo componente de base na representação proposta. A identificação das entidades e das proposições é levado a cabo no passo de análise sintático-semântica. Nossa análise das sentenças produz uma proposição para cada elemento predicativo e uma entidade para cada sintagma nominal da sentença.

5.1.1 Referência e co-referência

Cada entidade de um texto será representada na linguagem de representação proposta por um termo da forma (5.1)

$$p(\text{entidade}_i, \text{argumento}_2, \dots, \text{argumento}_n) \quad (5.1)$$

onde: (i) *entidade_i* é um índice que identifica de forma única o referente da expressão lingüística utilizada no texto para referenciá-lo; e (ii) *argumento_k* representa cada modificador utilizado na expressão lingüística. O índice deve ser obtido a partir do processo de interpretação das frases nominais. A análise sintático-semântica das frases nominais deve fornecer uma expressão de busca do referente e, portanto, do índice no banco de conhecimentos e, a partir desse índice, deve ser construída a expressão (5.1).

A partir da análise dos sumários notamos que existem expressões lingüísticas que têm poder de referência próprio, tais como os nomes próprios e algumas expressões que designam entidades reais ou conceituais pertencentes ao conhecimento partilhado pelo escritor e pelo leitor. Isto sugere que deve existir um conjunto de entidades identificáveis a partir de expressões lingüísticas consideradas completas e que o processo de identificação das entidades corresponde ao gráfico apresentado na Figura 5.3, onde uma expressão lingüística é utilizada para selecionar um referente em um banco de conhecimentos.

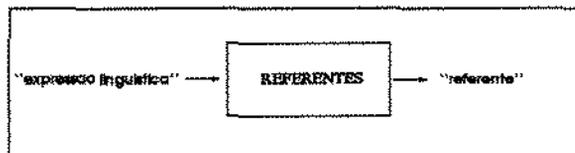


Figura 5.3: Obtenção de um referente

Por exemplo, consideremos um modelo de representação de conhecimento através de Redes Semânticas como o da Figura 5.4, onde vemos representada uma entidade com um índice “d1”. Essa entidade tem o nome “CALCULO NUMERICO” (arco NOME na representação) e é uma disciplina (arco E-UM na representação). Podemos supor que a análise da expressão lingüística “Cálculo Numérico” produz uma expressão de busca na rede que faz com que seja possível a obtenção da entidade com identificador “d1”. Dessa maneira toda vez que a expressão “Cálculo Numérico” aparecer em um texto o mesmo índice “d1” será obtido.

Por outro lado existem expressões que são introduzidas no contexto discursivo pela utilização de frases indefinidas. Então deve existir um mecanismo que possibilite a criação de uma entidade com todas as suas propriedades. Assim por exemplo na análise da expressão lingüística “um método” cria-se um referente novo na representação, como pode ser visto na Figura 5.5 (b), que não existia.

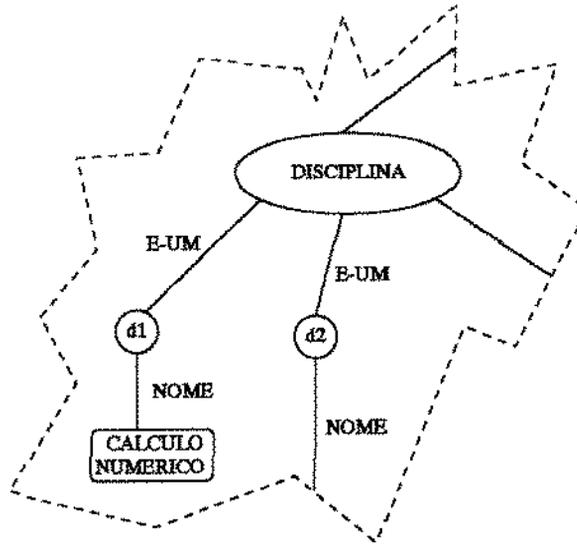


Figura 5.4: Uma representação

antes da análise da expressão lingüística, como representado na Figura 5.5 (a).

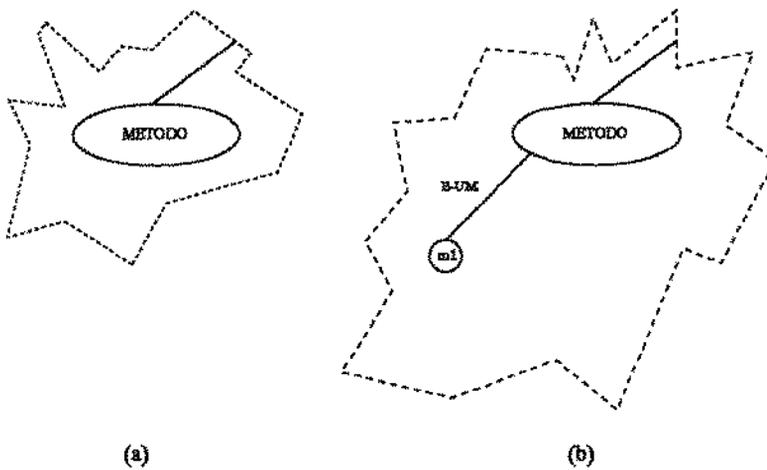


Figura 5.5: Criação de um referente

O modelo computacional deve ser capaz de lidar com as seguintes situações:

- (i) a *referência* a uma entidade pertencente ao domínio discursivo. Assim, por exemplo a expressão lingüística “Cálculo Numérico” se refere a uma entidade do domínio que tem a propriedade de ser uma disciplina;
- (ii) a *criação* de uma entidade. Por exemplo, a expressão lingüística “uma disciplina” introduz um novo referente que não existe no domínio discursivo. Esse referente deve ter todas as propriedades de uma disciplina;
- (iii) a *co-referência* a uma entidade previamente introduzida no texto. Por exemplo, a expressão lingüística “a disciplina” deve se referir a uma entidade previamente introduzida no texto;

- (iv) a *co-referência* a uma entidade não previamente introduzida no texto, mas que está conceitualmente vinculada a uma entidade previamente introduzida. Assim, a expressão lingüística “a metodologia de ensino” se referirá à metodologia de ensino de uma disciplina concreta somente se no contexto discursivo uma entidade com as propriedades de uma disciplina tiver sido introduzida previamente.

Para lidar com a *referência* é necessário que um conjunto de referentes exista no banco de conhecimentos antes da análise do texto; esses referentes representam as entidades conhecidas. Também necessitamos de um mecanismo para, a partir de uma expressão lingüística, identificar a identidade da entidade representada pela expressão.

Para suportar a *co-referência* por identidade é necessário que as entidades referenciadas ou criadas no decorrer da análise do texto estejam disponíveis para uma referência abreviada subsequente. Para a *co-referência* conceitual a uma entidade não introduzida explicitamente no texto, devemos ter um mecanismo para verificar relações entre entidades do domínio e outras entidades previamente introduzidas no texto.

5.1.2 Analisando expressões nominais

Consideremos o sumário da Figura 5.6. Apresentaremos a seguir as análises das frases nominais desse texto.

(1) Trocadores de calor compactos são elementos básicos e de alta eficiência. (2) Este trabalho apresenta um método simples (3) usado para o desenvolvimento, (4) teste e (5) análise térmica deste tipo de trocador. (6) Usando o método recomendado, (7) alunos de graduação podem estimar a queda da pressão e coeficientes de troca de calor (8) normalmente utilizados em engenharia térmica. [da Silveira Neto e Hernandez Mendoza 1988]

Figura 5.6: Frases nominais em um texto

Exemplo 30 Consideremos o caso de criação de uma referência em um texto, como o sintagma nominal “um método simples” na oração (2).

A análise sintática dessa expressão pode ser observada na Figura 5.7 e deve ser utilizada na construção da expressão na linguagem de representação.

O processamento da expressão deverá criar um novo referente no banco de conhecimentos com todas as propriedades de um método. Suponhamos que o índice criado a partir do processamento dessa expressão seja m_1 ; então o termo que representa essa entidade na linguagem de representação será (5.2).

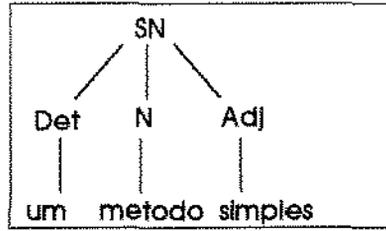


Figura 5.7: Sintagma nominal (1)

metodo(m1, qualificador = SIMPLES, nodet, mas, sin). (5.2)

O primeiro argumento é o identificador do referente, o segundo argumento é obtido a partir do adjetivo “simples” que modifica o núcleo do sintagma nominal e os outros argumentos são obtidos das informações sobre gênero e número do núcleo nominal e da operação de determinação (artigo indefinido).

Exemplo 31 Consideremos agora um caso de co-referência, como por exemplo o sintagma nominal “o método recomendado” da oração (6) do sumário da Figura 5.6. A sua análise pode ser observada na Figura 5.8.



Figura 5.8: Sintagma nominal (2)

Essa expressão é uma referência abreviada pois no léxico o núcleo dessa expressão representa uma entidade incompleta. A análise dessa expressão produz um processo de busca pelo antecedente baseado no predicador associado ao item lexical “método”. O antecedente dessa expressão deve ser uma entidade previamente introduzida no texto. A busca pelo antecedente finaliza com a obtenção do índice *m1* como referente da expressão lingüística; esse referente foi introduzido a partir da análise da expressão “um método simples” como descrito acima. O termo utilizado para representar neste caso a entidade *m1* é (5.3). O primeiro argumento é o índice que identifica a entidade e os outros argumentos são obtidos a partir do adjetivo “recomendado” que modifica o núcleo nominal, das informações de gênero e número e do determinante aplicado à expressão (artigo determinado).

metodo(m1, qualificador = RECOMENDADO, det, mas, sin). (5.3)

O fato dos termos (5.2) e (5.3) partilharem o mesmo índice indica que as expressões lingüísticas que os produziram são co-referentes.

Exemplo 32 Consideremos agora o sintagma nominal “trocadores de calor compactos” da oração (1) do sumário da Figura 5.6. A árvore de análise pode ser observada na Figura 5.9.

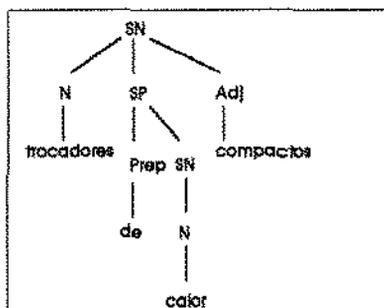


Figura 5.9: Sintagma nominal (3)

Suponhamos que o índice do referente dessa expressão seja $tc1$. Na linguagem de representação esse referente será representado pelo termo (5.4)

$$\begin{aligned} & trocador(tc1, \text{modificador} = CALOR, \\ & \text{qualificador} = COMPACTO, \text{none}, \text{mas}, \text{plu}). \end{aligned} \quad (5.4)$$

O primeiro argumento é o identificador da entidade representada e os outros argumentos são obtidos a partir do modificador nominal “calor”, do modificador adjetivo “compactos” e das informações sobre gênero e número do núcleo nominal e da operação de determinação (sem determinante).

Exemplo 33 No mesmo sumário aparece a expressão lingüística “este tipo de trocador” na oração (5); a árvore de análise é mostrada na Figura 5.10.

Essa expressão se refere a “trocadores de calor compactos” introduzidos previamente. A partir da análise dessa expressão obtemos o índice $tc1$ e construímos o termo (5.5).

$$tipo(tc1, \text{modificador} = TROCADOR, \text{dem1}, \text{mas}, \text{sin}). \quad (5.5)$$

Os termos (5.4) e (5.5) representam expressões lingüísticas co-referentes.

5.1.3 Proposições e relações entre proposições

Cada elemento predicativo tem definido no léxico da língua um conjunto de posições argumentais [Mateus et al. 1983] que são realizadas sintaticamente na superfície lingüística. Tendo como base essa informação tomada do léxico torna-se possível o cálculo das proposições. A partir de uma sentença do sumário obtemos

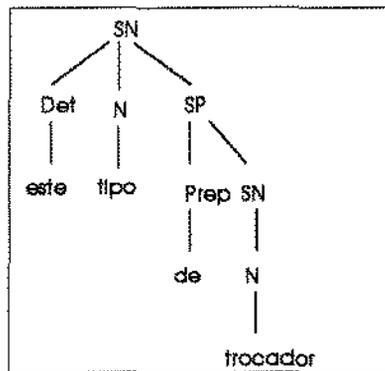


Figura 5.10: Sintagma nominal (4)

então uma ou mais proposições. Podemos distinguir entre proposições simples, que relacionam entidades, e proposições compostas, que relacionam entidades e também proposições. As construções complexas representam relações de tipo lógico-semânticas. A seguir apresentamos exemplos de casos analisados¹.

Consideremos o caso de uma sentença simples de um sumário:

Exemplo 34 A sentença “O trabalho apresenta uma experiência de laboratório” produz uma proposição derivada a partir do predicativo “apresenta”, que relaciona duas entidades i_1 “trabalho” e i_2 “experiência”.

$$\text{apresentar}(i_1, i_2) \quad (5.6)$$

Obtivemos esta representação a partir da análise sintático-semântica da sentença. Na Figura 5.11 podemos observar o resultado da análise sintática da sentença em questão.

A expressão “o trabalho” é analisada como um sintagma nominal em posição de sujeito; a expressão “uma experiência de laboratório” é classificada como um sintagma nominal realizado na posição de objeto direto. O verbo “apresenta” tem definidas duas posições argumentais na sua estrutura interna; esses argumentos são realizados na superfície lingüística como sintagmas nominais. Então a interpretação semântica da sentença associa o predicador **apresentar** com os seus dois argumentos, para produzir a proposição (5.6) apresentada anteriormente.

Consideremos agora o caso de uma sentença complexa de um sumário:

Exemplo 35 A sentença “a engenharia depende das ciências mas não prescinde da criatividade” produz a relação de tipo lógico semântica *Contraste*, marcada

¹Nas representações que apresentamos alguns elementos da linguagem de representação são omitidos para facilitar a leitura.

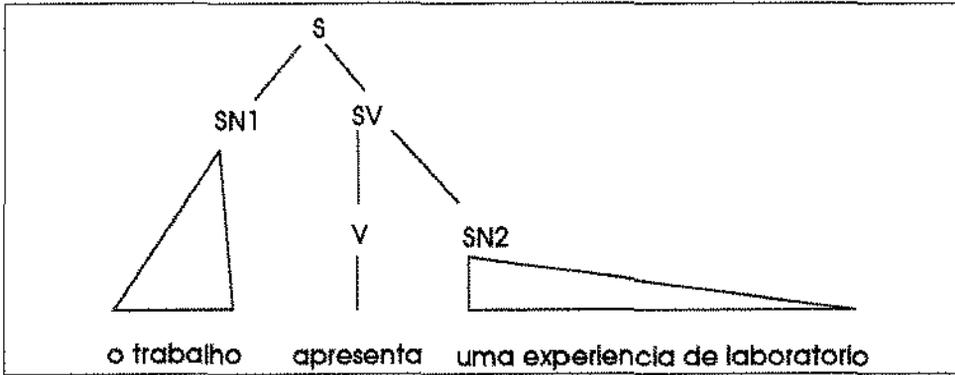


Figura 5.11: Análise sintática (1)

pela conjunção “mas”, que relaciona duas proposições simples que se correspondem com os predicativos “depende” e “prescinde”. As proposições são apresentadas a seguir, onde as constantes e_1 , e_2 e e_3 representam respectivamente as entidades “engenharia”, “ciências” e “criatividade”:

$$\text{depende}(e_1, e_2) \tag{5.7}$$

$$\text{não}(\text{prescindir}(e_1, e_3)) \tag{5.8}$$

$$\text{Contraste}(\text{depende}(e_1, e_2), \text{não}(\text{prescindir}(e_1, e_3))). \tag{5.9}$$

A árvore de análise da sentença pode ser observada na Figura 5.12.

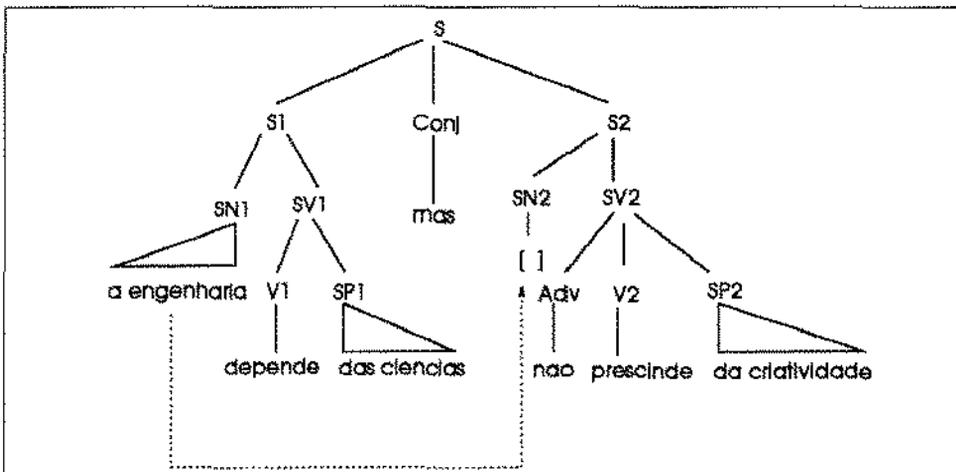


Figura 5.12: Análise sintática (2)

A primeira sentença contém o sintagma nominal “a engenharia” na posição de sujeito e o sintagma preposicionado “das ciencias” na posição de complemento preposicionado do verbo. O verbo “depende” tem definido no léxico duas posições

argumentais realizadas respectivamente como sintagma nominal e sintagma preposicionado. Portanto a interpretação dessa sentença produz a proposição (5.7). A segunda sentença não tem sujeito realizado; porém ele é interpretado como idêntico ao sujeito da primeira sentença. O sintagma preposicionado “da criatividade” é o complemento do verbo “prescinde” que tem no léxico duas posições argumentais. A análise semântica vincula o predicador **prescindir** obtido a partir do verbo “prescinde” com os seus argumentos para produzir a proposição (5.8). Finalmente a conjunção adversativa “mas” é interpretada como uma Relação Proposicional com duas posições argumentais, tal como apresentado em (5.9).

Consideremos agora o caso de uma relação de tipo coesiva no interior de uma sentença:

Exemplo 36 A sentença “O trabalho apresenta a disciplina Similitude em Engenharia ministrada na UNICAMP” contém dois predicativos “apresenta” e “ministrada”. Portanto a análise deverá produzir duas proposições vinculadas de alguma maneira na representação. Notamos que o segundo elemento predicativo “elabora” a entidade “Similitude em Engenharia” introduzida pelo primeiro, já que ambos os predicativos têm como argumento a mesma entidade “a disciplina Similitude em Engenharia”. Portanto esse tipo de construção será representada como uma elaboração que é interna à sentença. Abaixo apresentamos as proposições produzidas pela análise. As constantes t_1 , d_1 e u_1 representam respectivamente as entidades “trabalho”, “Similitude em Engenharia” e “UNICAMP”.

$$\text{apresentar}(t_1, d_1) \quad (5.10)$$

$$\text{ministrar}(d_1, u_1) \quad (5.11)$$

$$\text{Elaboração}(\text{apresentar}(t_1, d_1), \text{ministrar}(d_1, u_1)). \quad (5.12)$$

Na Figura 5.13 podemos observar a árvore de análise da sentença.

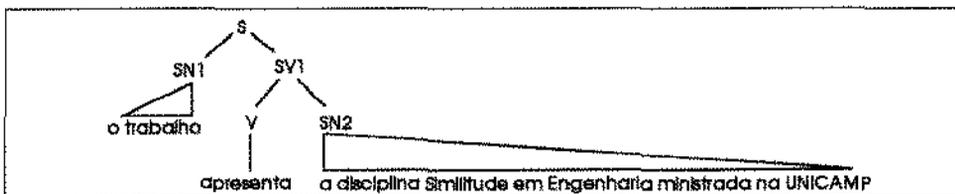


Figura 5.13: Análise sintática (3)

A sentença contém o sintagma nominal “o trabalho” realizado em posição de sujeito e o sintagma nominal “a disciplina Similitude em Engenharia ministrada na UNICAMP”, realizado em posição de complemento do verbo “apresenta”. O sintagma nominal SN2 contém o particípio “ministrada” que tem definido no léxico duas posições; por isso a análise semântica desse elemento predicativo produz a proposição (5.11). O verbo “apresentar” tem duas posições definidas no

léxico e então a análise semântica desse elemento produz a proposição (5.10). Finalmente, o fato das duas proposições conterem a mesma entidade faz com que seja também produzida a relação (5.12).

Consideremos agora uma justaposição de sentenças:

Exemplo 37 Na seqüência “O trabalho apresenta a disciplina Similitude em Engenharia. Esta disciplina visa proporcionar aos alunos fundamentos da teoria de modelos”, a primeira sentença contém apenas o predicativo “apresenta”; entretanto a segunda tem dois predicativos “visa” e “proporcionar”. Portanto a primeira sentença produz uma proposição simples a partir do predicativo “apresenta” e a segunda uma proposição complexa a partir do predicativo “visa”. As representações são apresentadas abaixo. As constantes a_1 e f_1 representam respectivamente as entidades “alunos” e “fundamentos da teoria de modelos”.

$$\text{apresentar}(t_1, d_1) \quad (5.13)$$

$$\text{proporcionar}(a_1, f_1) \quad (5.14)$$

$$\text{objetivo}(d_1, \text{proporcionar}(a_1, f_1)) \quad (5.15)$$

$$\text{Elaboração}(\text{apresentar}(t_1, d_1), \text{objetivo}(d_1, \text{proporcionar}(a_1, f_1))) \quad (5.16)$$

Na Figura 5.14 podemos observar a análise das duas sentenças da seqüência.

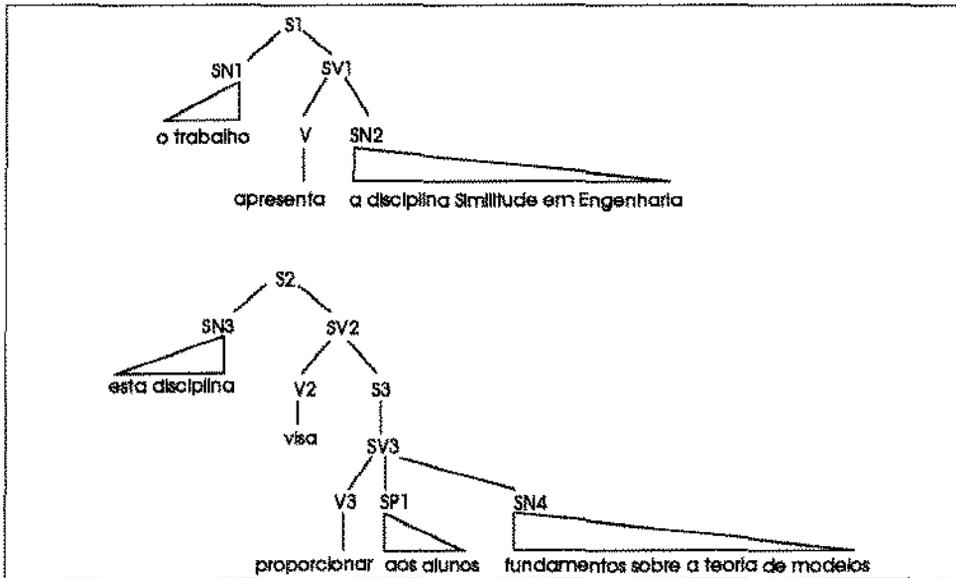


Figura 5.14: Análise sintática (4)

Na primeira sentença a expressão “o trabalho” é analisado como um sintagma nominal em posição de sujeito e a expressão “a disciplina Similitude em Engenharia” como um sintagma nominal em posição de complemento do verbo

“apresenta”. O verbo “apresenta” tem dois argumentos na sua estrutura interna que são realizados como sintagmas nominais; portanto a análise semântica dessa sentença associa o predicador **apresentar** obtido do verbo “apresenta” com os seus dois argumentos para produzir a proposição (5.13).

Na segunda sentença a expressão “esta disciplina” é um sintagma nominal em posição de sujeito. O sintagma verbal da sentença contém o verbo “visa” e um complemento realizado como frase infinitiva. A frase infinitiva contém o verbo no infinitivo “proporcionar” e dois complementos “aos alunos” (sintagma preposicionado) e “fundamentos sobre a teoria de modelos” (sintagma nominal). O verbo “proporcionar” tem definidos dois argumentos; então a análise dessa frase produz a proposição (5.14). O verbo “visa” tem dois argumentos na sua estrutura interna, realizados respectivamente como sintagma nominal e frase infinitiva; portanto a análise semântica da sentença associa o predicador **objetivo**, obtido do léxico a partir do verbo “visa”, com os seus dois complementos para obter a proposição (5.15).

Finalmente, como as proposições (5.13) e (5.15) partilham o mesmo argumento elas são vinculadas por uma relação de elaboração para obter a representação (5.16).

5.2 Identificação dos segmentos

Uma das dificuldades da análise textual está na determinação dos segmentos textuais. Dada uma seqüência textual de sentenças $S_1, \dots, S_k, \dots, S_n$, quando a sentença S_k está sendo processada deve-se determinar:

- (i) se S_k pertence ao segmento de texto sendo construído; ou
- (ii) se S_k é o começo de um novo segmento de texto.

Para resolver esse problema, [Morris e Hirst 1991] utilizam o conceito de *Coesão Lexical* dado por [Halliday e Hasan 1976]. Nesta abordagem, cadeias de palavras coesivamente vinculadas são identificadas utilizando-se relações entre palavras. Segundo estes autores, existe uma tendência a encontrar a finalização de um segmento quando uma cadeia de palavras finaliza. Alguns autores utilizam marcadores discursivos [Hirschberg e Litman 1993]; estes marcadores são a principal indicação do limite de um segmento e também da relação que se estabelece entre os segmentos vizinhos. Esse marcadores são utilizados no discurso no nível sentencial e no nível discursivo; portanto são ambíguos, fazendo-se necessário um estudo aprofundado para a determinação do seu uso no contexto [Litman 1994, Siegel e McKeown 1994]. [Pimenta Rodrigues e Pereira Lopes 1992] utilizam informação temporal para a determinação dos limites dos segmentos na sua estrutura do discurso.

Na nossa abordagem consideramos duas informações como relevantes na determinação dos segmentos do sumário. O primeiro é o uso das categoria sintática

tempo associado aos verbos. A diferenciação *passado*, *presente* é o principal indicador de uma mudança de segmento em qualquer tipo de texto [Grosz e Sidner 1986]. Essas informações estão presentes nos itens verbais do léxico da língua. No caso dos sumários essa diferenciação na utilização dos tempos verbais está relacionada com uma situação pragmática. Segundo as especificações sobre sumários feitas pelos analistas de documentação, o tempo passado é utilizado para descrever o trabalho científico; isto está relacionado com o fato do trabalho científico ter sido realizado antes da escrita do relatório. A categoria linguística tempo passado é, portanto, o principal indicador para determinar informação de tipo *resultado*, *metodologia* e *experimento*, todos indicadores do trabalho científico. O tempo presente é utilizado para indicar informações sobre o próprio relatório científico; portanto é uma marca dos indicadores de objetivos e conclusões do trabalho. A diferenciação passado/presente foi verificada nos textos analisados e portanto é utilizada na identificação dos segmentos.

Como exemplo consideremos o sumário da Figura 5.15. Podemos comprovar que o tempo presente é utilizado nas orações (1) e (2) que apresentam informação de tipo contextual. Na oração (3) é utilizado o tempo passado e essa oração indica informação sobre o trabalho científico. Finalmente a oração (4), no tempo presente, apresenta informação sobre os objetivos do artigo.

(1) *Os programas de avaliação de docentes pelos alunos vêm sendo utilizados nos Estados Unidos desde a década de quarenta.* (2) *Este tipo de programa não é muito comum nas universidades brasileiras.* (3) *No Departamento de Engenharia Mecânica da UNICAMP foram feitas algumas tentativas de implantação destes programas.* (4) *O presente trabalho descreve a experiência da aplicação destes programas no Departamento de Engenharia Mecânica da UNICAMP.* [Milanez 1987]

Figura 5.15: Exemplo de Sumário (1)

Entretanto a informação temporal pode indicar mais do que um tipo de segmento; portanto é necessário utilizar outras sinais para a correta identificação dos segmentos.

O segundo tipo de informação utilizada na determinação dos segmentos são os predicadores e as expressões nominais. Analisando os textos dos sumários e de acordo com as especificações apresentadas em [Jordan 1991], expressões típicas sobre objetivos foram verificadas no corpus de sumários. Expressões tais como “o propósito deste trabalho”, “o trabalho tem como objetivos”, “o trabalho visa” são marcadores dos objetivos do artigo ou do trabalho. As referências explícitas aos autores do artigo mediante a expressão “os autores” e as referências explícitas ao artigo mediante as expressões “o artigo” e “o trabalho” são indicadores dos tópicos que serão apresentados no artigo e portanto são considerados como mar-

cadres dos objetivos do artigo. Esta informação foi também verificada nos textos analisados. No sumário da Figura 5.15, podemos verificar esse fato. Na oração (4) a expressão “o presente trabalho” é uma referência ao artigo. No sumário da Figura 5.16, na oração (1) a expressão “o presente artigo” é também uma referência ao artigo.

(1) O presente artigo relata a metodologia de ensino de Projetos de Indústrias Químicas, (2) usada no Departamento de Engenharia Química do “University College” de Londres, Inglaterra, em 1985. (3) São mostrados os resultados obtidos naquela universidade e (4) comparados com os alcançados no Curso de Pós-Graduação de Projetos de Indústrias Petroquímicas, (5) ministrado no Departamento de Engenharia Química da Escola Politécnica da Universidade de São Paulo, em 1986. (6) São apresentadas algumas limitações e os méritos dessa metodologia que (7) são a criatividade, a relevância, as equipes de trabalho, o gerenciamento, a flexibilidade e a responsabilidade. [de Sá Weyne 1987]

Figura 5.16: Exemplo de Sumário (2)

Consideremos o sumário da Figura 5.17. Neste texto a expressão definida “os autores” na oração (4), refere-se aos autores do artigo; essa referência portanto indica que os objetivos do artigo estão sendo apresentados no segmento.

Um segmento de tipo contextual pode ser determinado pela ocorrência de alguns itens lexicais identificados com um problema ou uma necessidade de pesquisa. Quando não existem marcas explícitas, o contexto é determinado pela sua ocorrência no começo do texto do sumário, segundo as especificações colocadas por [Weissberg e Buker 1990] e segundo a verificação dos textos analisados. Um segmento de tipo metodologia pode ser determinado pela ocorrência de informação acerca do procedimento utilizado, dos materiais utilizados na pesquisa e das teorias empregadas. As conclusões apresentam em geral limitações e vantagens do trabalho apresentado assim como recomendações e sugestões.

Uma das hipóteses adotadas neste trabalho é a de segmentos textuais não encaixados. Um sumário é uma seqüência linear de segmentos; ele pode ser entendido como um subconjunto ordenado do conjunto de *Categorias Informacionais* adotado. Assim, por exemplo, o sumário da Figura 5.15 pode ser abstraído pela expressão (5.17)

$$\text{Sumário} = \text{Contexto.Experimento.Objetivo} \quad (5.17)$$

(1) *Constata-se hoje uma deficiência generalizada no ensino e conseqüente aprendizado de Desenho, na escola brasileira, em todos os níveis. (2) Fazendo-se uma análise do problema, (3) se verifica que em épocas anteriores à reforma do ensino instituída pela Lei 5.692/71, o Desenho tinha uma importância e um destaque bem acima da situação atual. (4) Os autores, após fazerem uma análise aprofundada da legislação existente sobre o assunto, principalmente após (5) a promulgação daquela Lei e de regulamentações posteriores, (6) partem para um estudo visando (7) encontrar soluções que (8) permitam a restituição do ensino de Desenho aos níveis do desenvolvimento tecnológico e industrial brasileiro. [Toropy e Aita 1987]*

Figura 5.17: Exemplo de Sumário (3)

A seqüência não pode ter elementos repetidos. Com essa hipótese a finalização do processamento de um segmento indica o começo de um segmento cuja categoria informacional não pertence à seqüência que está sendo construída.

A segunda hipótese adotada é que existem elementos estereotípicos para a identificação dos segmentos. Uma classificação de elementos típicos encontrados nos segmentos pode ser observada na Figura 5.18.

Exemplo 38 *Consideremos por exemplo a classificação da sentença (4) do sumário da Figura 5.15. A análise dessa sentença produz o conjunto de termos (5.18)*

$$(4) = \{\text{descrever}(t_1, e_1), \text{trabalho}(t_1, \dots), \dots\} \quad (5.18)$$

A classificação dessa sentença como pertencente a um segmento de tipo Objetivo procede como segue: (i) a sentença contém o predicador descrever que é um dos predicados marcadores de Objetivos; (ii) a sentença contém a entidade t_1 que representa o artigo; (iii) a entidade t_1 é argumento do predicado descrever. Portanto essa sentença é classificada como Objetivo.

5.3 Arquitetura do Analisador Textual

Nas Figuras 5.19 e 5.20 apresentamos a arquitetura detalhada do analisador textual proposto. Um protótipo experimental do sistema foi desenvolvido na linguagem de programação Prolog por considerá-la adequada para a análise das línguas naturais.

O analisador a nível sentencial (Figura 5.19) é composto dos seguintes processadores e dados:

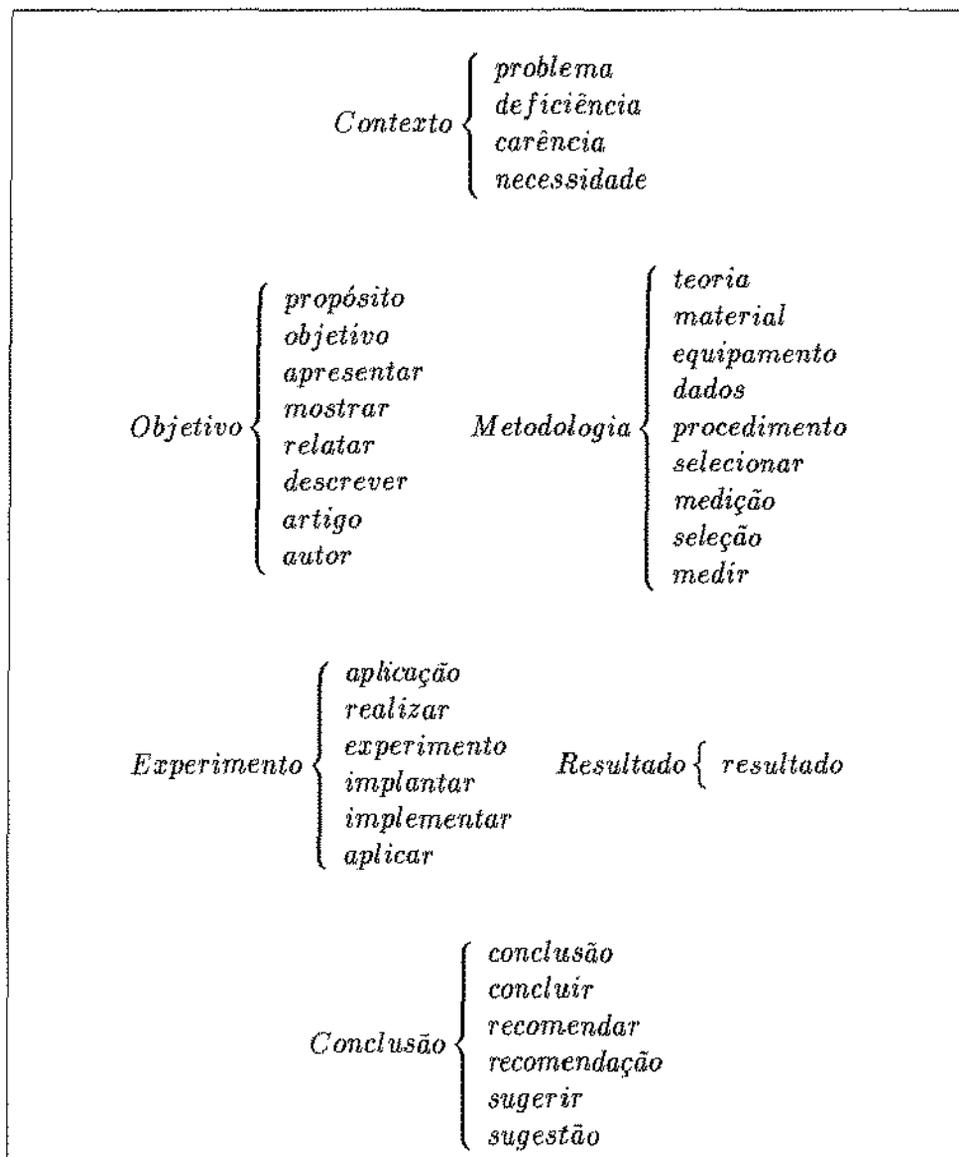


Figura 5.18: Marcadores de segmentos no corpus

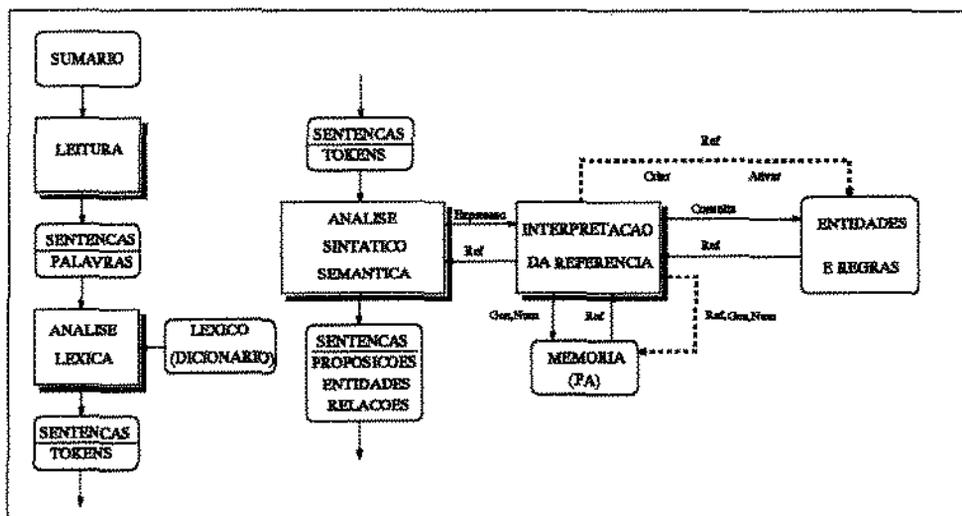


Figura 5.19: Arquitetura do analisador sentencial

- (i) *Sumário:*
esta estrutura consiste de uma seqüência de sentenças em língua portuguesa. Cada sentença é uma seqüência de palavras.
- (ii) *Leitura:*
este processo tem como entrada um sumário e produz por cada sentença do sumário uma lista de palavras.
- (iii) *Análise Léxica:*
este processo transforma uma lista de palavras em língua portuguesa em uma lista de “tokens” que contém as informações necessárias para o passo de análise sintático-semântica. A busca no dicionário é realizada utilizando-se a palavra completa.
- (iv) *Léxico ou Dicionário:*
esta estrutura contém para cada palavra em língua portuguesa, um token que codifica as informações sintáticas e semânticas associadas à palavra.
- (v) *Análise Sintático-Semântica:*
este processo tem como entrada uma lista de tokens e produz proposições, entidades e relações proposicionais a partir das regras da gramática e das regras de interpretação semântica associadas à gramática. Cada sintagma nominal é transformado em um termo que consiste de um predicador associado ao núcleo do sintagma nominal e argumentos associados aos modificadores do núcleo. A partir da interpretação da referência obtém-se um identificador da entidade a qual a expressão nominal se refere; esse identificador também é incorporado ao termo construído. Cada elemento predicativo é transformado em uma proposição construída a partir de um predicador associado ao elemento predicativo e argumentos obtidos a partir da análise dos complementos do predicativo. Cada proposição assim construída tem

associado um identificador incorporado à proposição. Sentenças com mais de um elemento predicativo serão vinculadas neste estágio utilizando-se relações proposicionais; neste caso um termo associado à relação também será construído. A relação utilizada para a vinculação depende da regra da gramática utilizada para a análise.

(vi) *Interpretação da Referência:*

este processo identifica a entidade a qual uma expressão nominal se refere. A partir de uma expressão nominal produz-se uma consulta ao banco de conhecimentos que contém entidades e regras de vinculação. O banco de conhecimentos responderá a essa consulta com o identificador da entidade. As expressões pronominais são buscadas em uma memória de pilha (PA) utilizando-se informações de gênero e número. A memória responderá a essa busca com o identificador do antecedente do pronome.

(vii) *Entidades e Regras:*

este banco de conhecimentos contém codificadas entidades e regras de vinculação entre entidades para a correta obtenção do referente de uma expressão nominal. Esta estrutura é atualizada no decorrer da análise para possibilitar tanto referências abreviadas como referências conceituais. As atualizações consistem tanto de incorporações de novas entidades ao banco de conhecimentos, como de ativações de entidades referenciadas no decorrer da análise do texto.

(viii) *Memória (PA):*

esta estrutura de memória de pilha contém referentes associados com os seus traços de gênero e número. A pilha é atualizada pelos referentes produzidos a partir da interpretação da referência; apenas são armazenados os traços de gênero e número associados com os referentes. A pilha é consultada utilizando-se informação sobre gênero e número.

(ix) *Proposições, entidades e relações:*

esta estrutura é um conjunto que contém os termos que representam o conteúdo proposicional da sentença: entidades, proposições e relações proposicionais codificadas como predicadores e argumentos.

O analisador a nível textual (Figura 5.20) é composto dos seguintes processadores e dados:

(i) *Classificação:*

este processo determina a categoria informacional de uma sentença baseando-se nos predicadores utilizados na sentença.

(ii) *Categorias:*

esta estrutura contém, para cada categoria informacional, os predicadores marcadores dessa categoria.

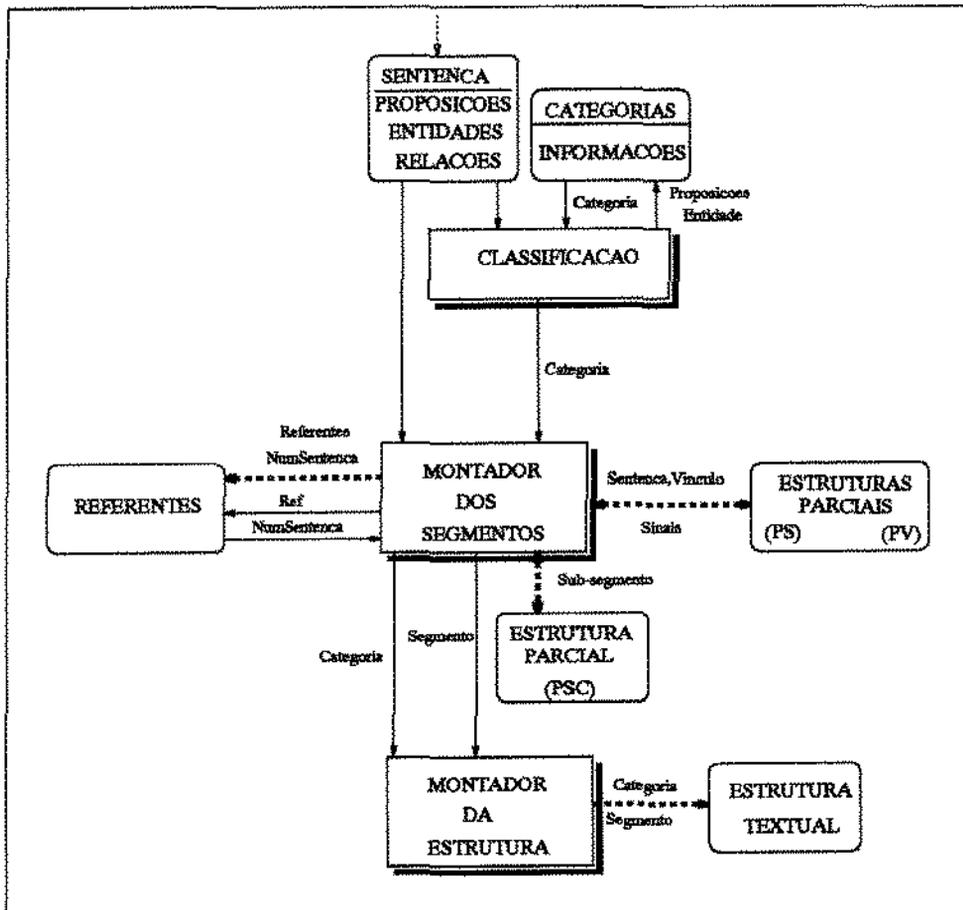


Figura 5.20: Arquitetura do montador da estrutura textual

(iii) *Referentes:*

nesta estrutura as entidades do sumário estão associadas à sentença na qual foram pela primeira vez introduzidas; esta informação é utilizada para vincular sentenças utilizando o conceito de coesão. A estrutura é consultada utilizando-se o identificador da entidade.

(iv) *Montador dos Segmentos:*

este processo vincula sentenças adjacentes que foram classificadas como pertencentes a mesma categoria. A vinculação é feita a partir da informação sobre os referentes. A primeira sentença de um segmento é armazenada na pilha PS (vazia); no processamento subsequente, uma sentença S_k pertencente ao segmento sendo processado será armazenada na pilha PS se S_k apresentar um vínculo coesivo com o elemento no topo de PS. Nesse caso a relação entre S_k e o elemento no topo de PS será refletida na pilha PV. Se S_k não apresentar vínculo com o topo de PS, uma redução será feita na pilha PS baseada nos vínculos de PV. As sentenças reduzidas serão vinculadas por relações de elaboração e paralelo para produzir um segmento que será armazenado na pilha PS.

(v) *Estruturas Parciais:*

a pilha PS contém um conjunto de sentenças com vínculos coesivos, onde cada elemento da pilha é um segmento de texto (uma sentença ou várias sentenças vinculadas por relações); a pilha PV contém o tipo de vinculação entre os elementos de PS.

(vi) *Estrutura Parcial:*

os segmentos reduzidos a partir de PS são passados para a estrutura PSC, onde ficarão até o final do processamento da categoria atual. Quando uma mudança de segmento acontecer, os segmentos de PSC serão vinculados uns aos outros por uma relação de seqüência para formar um único segmento de texto.

(vii) *Montador da Estrutura:*

neste processo a estrutura textual é construída a partir de um segmento obtido do montador de segmentos e dos segmentos processados previamente. Concatena-se o segmento atual aos segmentos processados previamente.

(viii) *Estrutura Textual:*

esta estrutura contém a representação final do sumário processado.

5.3.1 Iniciando a análise

O texto do sumário está contido em um arquivo `ascii`; cada registro do arquivo contém uma sentença representada por uma seqüência de palavras separadas por espaços em branco e que finaliza com um ponto. O texto finaliza com uma marca de fim de sumário. Na Figura 5.21 é apresentado a imagem de um arquivo que

contém o texto de um sumário analisado.

```

os trocadores de calor sao elementos basicos.
o trabalho apresenta um metodo usado para o teste deste tipo
de trocador.
usando o metodo recomendado os alunos podem estimar a
queda da pressao.
_ (→ marca de fim do sumário)

```

Figura 5.21: Arquivo contendo um sumário a ser analisado

O processo de análise completa do sumário é implementado pelo predicado Prolog *parsear_sumario(Nome_Arquivo)*, onde *Nome_Arquivo* é o nome do arquivo que contém o sumário a ser analisado.

O processo de leitura percorre o arquivo de entrada até a marca de fim de sumário. Uma palavra é reconhecida como uma seqüência de letras entre dois brancos ou delimitada pelo fim da sentença. Este processo é implementado pelo predicado Prolog *ler_sentencas(I)*, onde *I* é o número da sentença lida. Para cada registro no arquivo de entrada é produzida uma lista de átomos Prolog, onde cada átomo na lista representa uma palavra da língua portuguesa.

Os tokens que contém as informações sintático-semânticas necessárias para a análise das sentenças são codificados com os termos da Figura 5.22.

```

n(Nome, Tipo, Slots, Completude, Semantica, Genero, Numero)
  det(Determinante, Semantica, Genero, Numero)
    pronome(Pronome, Genero, Numero)
nome_proprio(Nome, Semantica1, Semantica2, Genero, Numero)
  prep(Preposicao)
v(Verbo, Slots, Semantica, Finitude, Numero, Tempo, Modo, Aspecto)
  contracao(Contracao, Preposicao, Artigo)
  adj(Adjetivo, Semantica, Genero, Numero)
  predicativo(Predicativo, Slots, Semantica, Genero, Numero)
    adv(Adverbio, Tipo, Semantica)
      coma(Coma)
    conjuncao(Conjuncao)
  coordenacao(Coordenacao, Tipo)
  relativo(Relativo)

```

Figura 5.22: Codificação dos tokens

O predicado Prolog *dicionario* incorpora esses termos Prolog a um dicionário que consiste de um conjunto de cláusulas codificadas como *dicionario(Palavra, Token)*, onde *Palavra* é um átomo que representa a palavra e *Token* é o token associado a essa palavra.

A lista de átomos produzida pelo processo de leitura é transformada em uma lista de tokens; esse processo de análise léxica é implementado pelo predicado Prolog *look_up(Lista_Palavras, Lista_Tokens)*, onde *Listas_Palavras* é uma lista de átomos e *Listas_Tokens* é a lista de tokens produzida pela análise. Cada átomo (palavra) é buscado no dicionário que contém as formas completas das palavras. Apenas as contrações sofrem um processo de análise morfológica; a contração é dividida em suas componentes e dois termos são produzidos, correspondendo à preposição e ao artigo que foram contraídos.

Exemplo 39 *Uma sentença no arquivo de entrada como*

o trabalho apresenta um metodo simples.

é transformada na lista Prolog

[o, trabalho, apresenta, um, metodo, simples]

e processada pelo analisador lexical para produzir a lista Prolog:

*{det(o,...), n(trabalho,...), v(apresenta,...), det(um,...), n(metodo,...),
adj(simples,...)}*

5.3.2 Análise das sentenças

Para analisarmos as sentenças de um subconjunto da língua portuguesa é necessário definirmos uma gramática (G) que descreva as formas superficiais aceitáveis e um algoritmo de análise tal que, a partir das regras da gramática e de uma sentença de entrada ($Sent$), decida se a sentença é válida de acordo com a gramática ($Sent \in L(G)$). Além de testarmos a pertinência é necessário construirmos uma representação de significado da sentença que possa ser utilizada no processamento subsequente.

As *Gramáticas de Cláusulas Definidas (DCGs)* [Pereira e Warren 1980] tornam possível a escrita de regras para a descrição de frases aceitáveis de uma língua. As regras são compiladas automaticamente pelo intérprete Prolog para produzir um programa em lógica (um conjunto de cláusulas Prolog). Esse programa representa um analisador derivado da estratégia de busca de Prolog. A

interface com o analisador produzido desta forma pode ser realizada diretamente utilizando o axioma da gramática. Consideremos, por exemplo, a gramática $G = \{N, T, P, \textit{Sentenca}\}$ apresentada na Figura 5.23 que descreve dentre outras a sentença do português “o trabalho apresenta um método simples”.

```

N = {Sentenca, Sn, Sv, N, V, Adj, Det}
T = {o, um, trabalho, método, simples, apresenta}
P =
Sentenca → Sn Sv
Sn → Det N
Sn → Det N Adj
Sv → V Sn
Det → o
Det → um
N → trabalho
N → método
Adj → simples
V → apresenta

```

Figura 5.23: Uma gramática

Essa gramática pode ser re-escrita em DCG de forma quase direta, como podemos observar na Figura 5.24

```

sentenca → sn, sv
sn → det, n
sn → det, n, adj
sv → v, sn
det → [o]
det → [um]
n → [trabalho]
n → [metodo]
adj → [simples]
v → [apresenta]

```

Figura 5.24: Uma gramática em DCG

Os símbolos não terminais são transformados em símbolos Prolog, os elementos da parte direita das regras são separados por vírgula e os símbolos terminais são re-escritos entre chaves na notação de lista de Prolog. O compilador DCG transforma a gramática em um programa Prolog onde cada não terminal da gramática com n argumentos é transformado em um predicado Prolog com $n + 2$

argumentos. Os dois argumentos acrescentados indicam a sentença a ser analisada pelo predicado e a parte final da sentença que ficou sem analisar. Se uma consulta como

$$\text{simbolo}(S_0, S_f)$$

resultar em um sucesso significa que a cadeia $S_0 - S_f$ é uma frase de tipo *simbolo*. Dado que na gramática apresentada os não terminais não possuem argumentos, eles serão transformados em predicados binários. Para analisar a sentença considerada devemos transformá-la em uma lista de átomos Prolog e realizar uma consulta utilizando o axioma da gramática:

$$\text{sentenca}([o, \text{trabalho}, \text{apresenta}, \text{um}, \text{metodo}, \text{simples}], []).$$

Com a gramática considerada apenas podemos verificar a pertinência de uma expressão na linguagem gerada pela gramática. Porém o formalismo faz com que seja possível a incorporação de parâmetros aos não terminais, assim como que cada regra seja acrescida de chamadas a predicados Prolog para interpretação semântica. Com o mecanismo de passagem de parâmetros podemos construir a árvore de análise de uma sentença modificando a gramática como aparece na Figura 5.25.

$\begin{aligned} \text{sentenca}(\text{sent}(Asn, Asv)) &\rightarrow \text{sn}(Asn), \text{sv}(Asv) \\ \text{sn}(\text{sn}(Adet, An)) &\rightarrow \text{det}(Adet), n(An) \\ \text{sn}(\text{sn}(Adet, An, Aadj)) &\rightarrow \text{det}(Adet), n(An), \text{adj}(Aadj) \\ \text{sv}(\text{sv}(Av, Asn)) &\rightarrow v(Av), \text{sn}(Asn) \\ \text{det}(\text{det}(o)) &\rightarrow [o] \\ \text{det}(\text{det}(\text{um})) &\rightarrow [\text{um}] \\ n(n(\text{trabalho})) &\rightarrow [\text{trabalho}] \\ n(n(\text{metodo})) &\rightarrow [\text{metodo}] \\ \text{adj}(\text{adj}(\text{simples})) &\rightarrow [\text{simples}] \\ v(v(\text{apresenta})) &\rightarrow [\text{apresenta}] \end{aligned}$
--

Figura 5.25: Uma gramática em DCG com parâmetros

A análise da sentença será feita agora utilizando a consulta:

$$\text{sentenca}(\text{Arvore}, [o, \text{trabalho}, \text{apresenta}, \text{um}, \text{metodo}, \text{simples}], [])$$

que produzirá a resposta:

$$\begin{aligned} \text{Arvore} = &\text{sent}(\text{sn}(\text{det}(o), n(\text{trabalho})), \\ &\text{sv}(v(\text{apresenta}), \text{sn}(\text{det}(\text{um}), n(\text{metodo}), \text{adj}(\text{simples})))) \end{aligned}$$

O mecanismo de passagem de parâmetros resulta apropriado para a verificação da concordância em gênero, número e pessoa entre os constituintes da sentença.

O formalismo *Slots Grammars* [McCord 1989] tem sido utilizado para a análise de línguas naturais. A idéia é que cada frase tem um elemento nuclear que deve especificar quais são os seus complementos e, desta maneira, a análise é guiada pela informação obtida do léxico sobre o quadro de subcategorização dos itens da língua. Assim no formalismo original uma lista de “slots” é acrescida a cada elemento do léxico indicando os complementos que acompanham o item na superfície lingüística. Uma lista de “slots” tem a forma

$$(Slot_1 : Marcador_1).(Slot_2 : Marcador_2)...(Slot_n : Marcador_n).nil$$

onde $Slot_i$ indica o nome do “slot” e $Marcador_i$ é o identificador da frase que preenche o i -ésimo “slot” na estrutura sintática da frase. Uma lista típica de “slots” para um verbo como o verbo “give” do inglês é

$$(subj : X).(obj : Y).(iobj : Z).nil$$

que indica que o verbo tem três argumentos na sua estrutura interna e que esses argumentos realizam as funções sintáticas de sujeito (subj), objeto (obj) e objeto indireto (iobj). Neste exemplo os slots utilizados representam funções gramaticais. Na Figura 5.26 apresentamos um exemplo simples de como uma regra para analisar um sintagma verbal é especificada no formalismo das *Modular Logic Grammars* [McCord 1986], que utiliza idéias de Slot Grammars.

$$\begin{aligned} vp(X) &\Rightarrow vhead(X, Slots) : postmods(Slots) \\ postmods((Slot : X).Slots) &\Rightarrow fill(Slot, X) : postmods(Slots) \\ postmods(nil) &\Rightarrow nil \\ fill(obj, X) &\Rightarrow np(X) \\ fill(iobj, X) &\Rightarrow +to : np(X) \end{aligned}$$

Figura 5.26: Uma Slot Grammar

Este conjunto de regras especificam que um sintagma verbal é composto por um núcleo verbal seguido de modificadores. Os modificadores são especificados pelo verbo.

Em nossa abordagem utilizamos DCGs acrescidas com idéias advindas de Slot Grammars incorporando a cada elemento da língua com valor predicativo (verbos, alguns adjetivos e alguns nomes) uma lista de “slots” que indicam o quadro de subcategorização do item (realização sintática) e o papel semântico do argumento (relação do argumento com o predador). Com essa informação os itens

nucleares de cada frase guiam a análise. Cada frase é analisada como contendo um elemento nuclear e um conjunto de complementos que se correspondem com a informação tomada do léxico. Cada slot é um par $[Papel, Tipo_Frase]$, onde *Papel* indica o papel semântico do argumento e *Tipo_Frase* indica a realização do argumento na superfície lingüística.

Assim, por exemplo, o item “formação” é considerado um nominal com valor predicativo. No léxico da língua ele tem especificado um complemento com papel semântico de *recipiente* e realizado sintaticamente como sintagma preposicionado com preposição “de”. Então, além da informação sintática e semântica, a entrada no dicionário para esse item conterà a informação

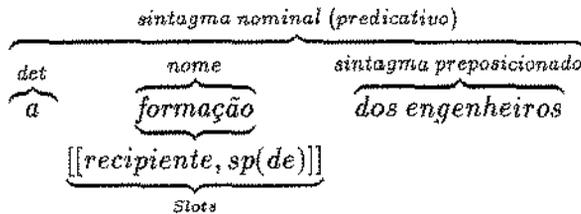
$[recipiente, sp(de)]$.

Para analisarmos um sintagma nominal como “a formação dos engenheiros” podemos utilizar a regra DCG da Figura 5.27, onde os argumentos para interpretação semântica foram omitidos.

$sn \rightarrow det, n(Slots), preencher(Slots).$
 $preencher(Slots) \rightarrow selecionar(Slots, Slot, Slots1),$
 $preencher_um(Slot),$
 $preencher(Slots1).$
 $preencher([]) \rightarrow [].$
 $preencher_um([Sem, sp(X)]) \rightarrow sp(X).$

Figura 5.27: Análise de um nominal com valor predicativo

Abaixo apresentamos um esquema de análise desse sintagma.



O processo de interpretação semântica deste tipo de construção cria um referente e constrói uma proposição que tem como predicador principal o item semântico associado ao núcleo e como argumentos os referentes produzidos a partir da análise dos complementos do núcleo. As entidades e proposições produzidas pela análise do sintagma são:

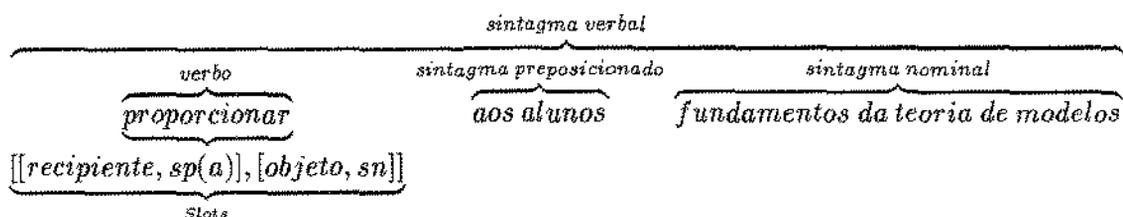
$Proposições = [formar(p_k, recipiente = e_j, det, fem, sin)]; e$

$Entidades = [engenheiro(e_j, det, mas, plu)].$

Para a análise dos sintagmas verbais utiliza-se uma regra similar. Assim, por exemplo, uma oração reduzida de infinitivo como “proporcionar aos alunos fundamentos da teoria de modelos” pode ser analisada nesta abordagem levando em consideração o fato do verbo “proporcionar” ter uma lista de dois “slots” definidos no léxico da língua, da forma:

$$[[\textit{recipiente}, sp(a)], [\textit{objeto}, sn]].$$

A análise da frase pode ser realizada de forma similar à anterior. Da análise dos complementos do verbo obtemos os referentes para a construção da representação da proposição. O esquema da análise é apresentado abaixo:

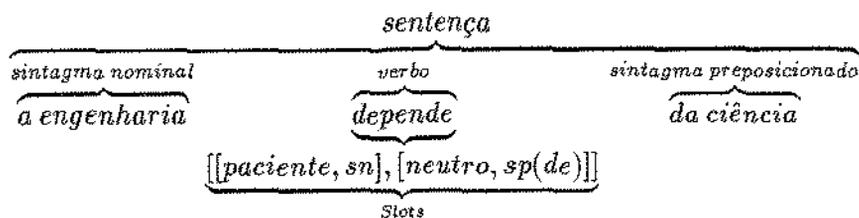


Como resultado da interpretação semântica deste sintagma obtemos os seguintes conjuntos de entidades e proposições:

Proposições = [*proporcionar*(*p_l*, *recipiente* = *e_p*, *objeto* = *e_h*, *inf*, *inf*, *inf*)];
e

Entidades = [*aluno*(*e_p*, *det*, *mas*, *plu*),
fundamento(*e_h*, *especificador* = *e_t*, *none*, *mas*, *plu*),
teoria(*e_t*, *modificador* = *modelo*, *det*, *fem*, *sin*)].

Para a análise de uma sentença completa também é utilizada a informação sobre os slots associados aos predicativos. Uma sentença simples é analisada como formada por um sintagma nominal e um sintagma verbal; a informação obtida a partir da análise do sintagma nominal é passada para o sintagma verbal para satisfazer a lista de slots associada ao verbo principal. Uma sentença simples como “a engenharia depende da ciência” será analisada segundo o seguinte esquema:



A interpretação desta sentença produzirá os seguintes conjuntos de entidades e proposições:

Proposições = [*depende*(*p_m*, *paciente* = *e_q*, *neutro* = *e_f*,
fin, *ind*, *acabado*)]; e

Entidades = [*engenharia*(*e_f*, *det*, *fem*, *sin*), *ciencia*(*e_f*, *det*, *fem*, *sin*)].

O processamento de cada sentença do sumário é realizado pelo predicado Prolog

parsear1(*Lista_Tokens*, *Arvore*, *Entidades*, *Proposicoes*, *Tempo*, *Aspecto*)

onde *Lista_Tokens* contém a lista de tokens a ser analisada, *Arvore* é a árvore produzida como resultado da análise, *Entidades* é uma lista de termos que representam as entidades produzidas pela análise da sentença, *Proposicoes* é uma lista de termos que representam as proposições e as relações proposicionais obtidas a partir da análise, *Tempo* é a informação sobre tempo do verbo principal e *Aspecto* é a informação aspectual do verbo principal. O predicado *parsear1* contém uma chamada ao predicado

s(*Tipo*, *Arvore*, *Variavel*, *Adianto*, *Entidades*, *Predicados*, *Tempo*, *Aspecto*)

encarregado da análise da sentença utilizando-se das regras DCGs. Existe uma cláusula para cada tipo de sentença.

Cada sintagma nominal é analisado pelo predicado Prolog

np(*Referente*, *Entidade*, *Arvore*, *Genero*, *Numero*,
Entidades, *Predicados*, *Arg*)

onde *Referente* é o identificador do referente do sintagma obtido a partir do processo de interpretação, *Entidade* é o termo que descreve a entidade descrita pelo sintagma nominal; *Arvore* é a árvore de análise do sintagma nominal, *Genero* é a informação de gênero do sintagma nominal, *Numero* é a informação de número do sintagma nominal, *Entidades* é uma lista que contém as entidades produzidas no decorrer da análise do sintagma, *Predicados* é uma lista que contém as proposições produzidas no decorrer da análise do sintagma e *Arg* indica se o sintagma ocupa a posição argumental de um predicativo.

Cada sintagma verbal é analisado pelo predicado Prolog

vp(*Finitude*, *Adelanto*, *Arvore*, *Variavel*, *Numero*, *Tempo*, *Aspecto*,
Entidades, *Predicados*)

onde *Finitude* indica o tipo de sintagma verbal (finito, infinitivo, gerundio, particípio), *Adelanto* contém informações sobre a frase que antecede ao sintagma verbal e que preenche uma posição argumental na estrutura do verbo, *Arvore* é arvore de análise do sintagma verbal, *Variavel* é o identificador associado com o verbo do sintagma verbal, *Numero* é a informação de número utilizada para concordância, *Tempo* é a informação sobre tempo associada ao sintagma, *Aspecto* é a informação aspectual associada ao sintagma, *Entidades* é uma lista das entidades produzidas a partir da análise do sintagma verbal e *Predicados* é uma lista de proposições produzidas a partir da análise do sintagma verbal.

Exemplo 40 A sentença:

usando o método os alunos podem estimar a queda da pressão.

será analisada como uma construção subordinada circunstancial por ter uma oração reduzida de gerundio. Abaixo apresentamos o esquema da análise desta sentença.

oração reduzida (gerundio)
sentença principal
usando o método
os alunos podem estimar a queda da pressão

e serão produzidos os seguintes conjuntos:

Entidades = [metodo(x_1 , det, mas, sin), aluno(e_1 , det, mas, plu), queda_pressao(e_2 , det, fem, sin)]

*Proposições = [usar(p_1 , objeto = x_1 , ger, ger, pro),
 poder(p_2 , agente = e_1 , fim = p_3 , fin, ind, fin),
 estimar(p_3 , objeto = e_2 , inf, inf, inf),
 ocasiao(p_4 , p_1 , p_2)]*

5.3.3 Resolvendo pronomes

Para a resolução dos pronomes de terceira pessoa utilizamos uma memória com organização de pilha. A escolha deste tipo de dado foi baseada na observação da utilização dos pronomes no corpus analisado. Um pronome é considerado uma anáfora; portanto o seu antecedente deverá ser uma entidade introduzida previamente à ocorrência do pronome no texto. Nos sumários analisados o antecedente do pronome é a entidade do discurso mais próxima e que concorda em traços sintáticos com o pronome. Dado que os pronomes apenas trazem instruções de conexão definidas no léxico da língua, são essas as informações que utilizaremos na sua resolução e que deverão estar presentes na memória.

Utilizamos uma pilha PA para o armazenamento dos possíveis antecedentes para um pronome e consideramos todas as entidades do discurso prévio como possíveis antecedentes. Cada elemento da pilha é uma 3-tupla cujos componentes são *índice*, *gênero* e *número* indicando o índice associado à entidade, seu gênero e número. As operações sobre a pilha são as usuais: *topo*, *empilhar* e *desempilhar*. Quando uma nova entidade é introduzida no texto é produzida uma expressão como (5.19) na linguagem de representação:

$$p(i_e, \dots, d, g, n). \quad (5.19)$$

Baseados nessa expressão atualizamos a pilha utilizando a operação (5.20):

$$PA \leftarrow empilhar(PA, \langle i_e, g, n \rangle) \quad (5.20)$$

O processo de atualização da pilha deve ser acrescido às regras de análise dos sintagmas nominais não pronominais.

Quando um pronome é encontrado no texto é produzida a expressão (5.21),

$$anafora(i_a, g_a, n_a) \quad (5.21)$$

onde i_a é um índice associado provisoriamente à anáfora, g_a é a informação de gênero e n_a é a informação de número.

O passo de busca do antecedente para o pronome baseia-se na especificação (5.22). É procurado o antecedente *referente*(X) na pilha PA até obter concordância em gênero e número:

$$X \in PA / genero(X) = g_a \wedge numero(X) = n_a \quad (5.22)$$

A pilha é implementada em Prolog como uma lista $[Tope|Resto]$. Uma pilha vazia é implementada como uma lista vazia e as operações sobre a pilha são definidas como:

$$\begin{aligned} &tope([Tope|Resto], Tope). \\ &empilhar(Elemento, Pilha, [Elemento|Pilha]). \\ &desempilhar([Tope|Resto], Resto). \end{aligned}$$

A 3-tupla é implementada em Prolog como uma lista $[Ref, Gen, Num]$ e as operações são definidas como:

$$\begin{aligned} &referente([Ref, Gen, Num], Ref) \\ &genero([Ref, Gen, Num], Gen) \\ &numero([Ref, Gen, Num], Num) \end{aligned}$$

O processo de interpretação dos pronomes é implementado pelo predicado Prolog *busca_ref*(*Genero*, *Numero*, *Referente*), onde *Genero* é o gênero do pronome, *Numero* é o número do pronome e *Referente* é o identificador do referente encontrado.

5.3.4 Interpretando frases nominais

Para a identificação do referente de uma expressão lingüística utilizamos um banco de conhecimentos Prolog que contem referentes para as entidades do domínio discursivo. A análise de um sintagma nominal pelas regras DCG definidas produz uma consulta ao banco de conhecimentos para a localização do referente da expressão. Os referentes são representados no banco de conhecimentos por cláusulas unitárias como

$$p(ref, \dots)$$

onde o referente *ref* pode ser localizado com uma consulta utilizando o predicado *p*.

Codificamos uma entidade como **Cálculo Numérico** utilizando as cláusulas

$$\begin{aligned} &disciplina(c_1) \\ nome(c_1, calculo_numerico) \end{aligned}$$

onde *c₁* e *calculo_numerico* são constantes e *disciplina* e *nome* são predicados que indicam respectivamente que *c₁* é uma disciplina e o nome de *c₁* é *calculo_numerico*. A análise do sintagma nominal “Cálculo Numérico” produz a expressão de busca

$$nome(X, calculo_numerico)$$

e o banco de conhecimentos responde a essa consulta com $X = c_1$. Uma vez que essa entidade foi referenciada pela utilização de um nome próprio, ela deve ficar ativa para uma referência abreviada subsequente. Isto é feito incorporando ao banco de conhecimentos a cláusula:

$$ativa(c_1).$$

Consideremos o caso do sintagma nominal “a disciplina”; esta expressão é considerada incompleta segundo a informação tomada do léxico. A análise dessa expressão produz a expressão de busca

$$disciplina(X), ativa(X)$$

no banco de conhecimentos, que finaliza com resultado $X = c_1$.

Consideremos agora uma entidade como o **plano de ensino de Cálculo Numérico**, que é uma entidade vinculada conceitualmente a **Cálculo Numérico**. Representamos essa entidade utilizando as cláusulas

$$\begin{aligned} & \textit{plano}(p(c1), \textit{ensino}) \\ & \textit{especificador}(p(c1), c1) \end{aligned}$$

onde $p(c1)$ é o termo que identifica a entidade, a primeira cláusula indica que $p(c1)$ é um **plano de ensino** e a segunda indica que as entidades com índices $p(c1)$ e $c1$ estão vinculadas. A análise da expressão lingüística “o plano de ensino de Cálculo Numérico” produz a expressão de busca no banco de conhecimentos

$$\textit{plano}(X, \textit{ensino}), \textit{especificador}(X, Y), \textit{nome}(Y, \textit{calculo_numerico})$$

cujo resultado é $X = p(c_1), Y = c_1$. O referente $p(c_1)$ deve ser marcado como ativo no banco de conhecimentos. Isto é feito incorporando ao banco de conhecimentos a cláusula:

$$\textit{ativa}(p(c_1)).$$

Consideremos agora a relação entre uma disciplina e o seu plano de ensino. Representamos esta relação no banco de conhecimentos pela regra

$$\textit{plano}(p(X), \textit{ensino}) : \neg \textit{disciplina}(X), \textit{ativa}(X).$$

que interpretamos como:

$p(X)$ é o plano de ensino de X se X é uma disciplina e X está ativa no banco de conhecimentos (foi referenciada previamente).

Assim, a análise de uma expressão como “o plano de ensino” produz a consulta ao banco de conhecimentos

$$\textit{plano}(X, \textit{ensino})$$

cuja resposta é $X = p(c_1)$.

Consideremos o caso de uma expressão indefinida como “um método” que cria um referente no banco de conhecimentos. Um índice m_1 é criado e as cláusulas

$$\begin{aligned} & \textit{metodo}(m_1) \\ & \textit{ativa}(m_1) \end{aligned}$$

são incorporadas ao banco. Uma expressão lingüística como “o método”, que é considerada incompleta, pode obter o seu referente m_1 com a consulta:

$$\text{metodo}(X), \text{ativa}(X).$$

A relação entre uma **disciplina** e o seu **ciclo de aulas** pode ser expressa pela regra

$$\text{ciclo}(c(X), \text{aula}) : -\text{disciplina}(X), \text{ativa}(X)$$

que interpretamos como

c(X) é o ciclo de aulas de X se X é uma disciplina e X está ativa no banco de conhecimentos no momento de realizarmos uma referência a c(X).

Também incorporamos as regras

$$\text{tipo}(X, \text{programa}) : -\text{programa}(X), \text{ativa}(X)$$

$$\text{tipo}(X, \text{trocaador}) : -\text{trocaador}(X), \text{ativa}(X)$$

que indicam respectivamente o conhecimento lingüístico sobre as expressões “tipo de programa” e “tipo de trocaador”:

X é um tipo de programa se X é um programa e X está ativo no momento de efetuarmos a referência;

e

X é um tipo de trocaador se X é um trocaador e X está ativo no momento de efetuarmos a referência.

Podemos concluir que cada regra de análise de um sintagma nominal não pronominal contém um processo de interpretação que gera uma busca no banco de conhecimentos que depende das operações de determinação e da completude do núcleo nominal. As regras para a análise dos sintagmas nominais com valor referencial contém uma chamada ao predicado Prolog

$$\text{interpretar}(\text{Entidade}, \text{Completude})$$

encarregado de construir a consulta ao banco de conhecimentos baseada no termo *Entidade* e no parâmetro *Completude*. A análise dos sintagmas nominais com operações de indeterminação não produzem consultas ao banco de conhecimentos, criam e ativam novos referentes.

5.3.5 Construindo a estrutura textual

Na construção da estrutura textual consideramos três processos: (i) a classificação das sentenças como pertencentes a um tipo de segmento ou categoria informacional; (ii) a vinculação das sentenças no segmento; e (iii) a vinculação dos segmentos na estrutura textual.

As sentenças são classificadas levando-se em consideração as especificações descritas na Seção 5.2. A escolha de uma Categoria Informacional para uma sentença depende dos predicadores e das entidades da sentença. Uma sentença que não pode ser classificada é considerada como pertencente ao segmento sendo processado. A classificação de uma sentença é realizada pelo predicado Prolog *classifica(I, Categoria)* encarregado de obter a categoria informacional *Categoria* para a sentença com número *I*.

A vinculação entre sentenças pertencentes a um segmento é realizada considerando vínculos coesivos obtidos a partir da resolução das referências. No decorrer da análise construímos uma estrutura textual que chamamos *segmento*. Um segmento é especificado como:

$$\text{Segmento} \left\{ \begin{array}{l} \text{Sentença} \\ \text{Seqüência}(\text{Segmento}_1, \dots, \text{Segmento}_n) \\ / \text{Segmento}_k \ 1 \leq k \leq n \text{ é um segmento} \\ \text{Elaboração}(\text{Segmento}_1, \text{Segmento}_2) \\ / \text{Segmento}_k \ k=1,2 \text{ é um segmento} \\ \text{Paralelo}(\text{Segmento}_1, \dots, \text{Segmento}_m) \\ / \text{Segmento}_k \ 1 \leq k \leq m \text{ é um segmento} \end{array} \right.$$

O algoritmo completo para a construção da estrutura é especificado como:

Seja CA a categoria informacional do segmento atual e seja S a sentença a ser processada.

- (i) Seja C a categoria informacional da sentença S;
- (ii) Se $C = CA$ a sentença S passa para o processo de montagem do segmento;
- (iii) Se $C \neq CA$
 - (a) o processamento do segmento associado a CA é finalizado;
 - (b) o segmento associado a CA passa para o montador da estrutura textual;
 - (c) $CA \leftarrow C$;
 - (d) S passa para o processo de montagem do segmento.
- (iv) Se não houver mais sentenças a serem processadas;
 - (a) o processamento do segmento associado a CA é finalizado;
 - (b) o segmento associado a CA passa para o montador da estrutura textual;
 - (c) fim do processamento.

Para o processo de *montagem do segmento* utilizamos os seguintes padrões observados nos textos analisados

$$\overbrace{S_1, S_2} \Rightarrow \text{Elaboração}(S_1, S_2) \quad (5.23)$$

que interpretamos como:

uma justaposição de uma sentença S_1 e um segmento de texto S_2 , tal que S_2 elabora uma entidade de S_1 , serão vinculados utilizando a relação de elaboração para obter $\text{Elaboração}(S_1, S_2)$;

$$\overbrace{S_1, \dots, S_n} \Rightarrow \text{Paralelo}(S_1, \dots, S_n) \quad (5.24)$$

que interpretamos como:

uma justaposição de n segmentos de texto S_1, S_2, \dots, S_n que elaboram uma entidade, serão vinculados utilizando a relação de paralelo para obter $\text{Paralelo}(S_1, S_2, \dots, S_n)$;

$$S_1, \dots, S_n \Rightarrow \text{Seqüência}(S_1, \dots, S_n) \quad (5.25)$$

que interpretamos como:

uma justaposição de segmentos de texto S_1, S_2, \dots, S_n sem elaborações serão vinculados utilizando a relação de seqüência para obter $\text{Seqüência}(S_1, S_2, \dots, S_n)$.

Utilizamos três pilhas para o processamento de um segmento: a pilha PS para o armazenamento de segmentos com vínculos coesivos no segmento atual; a pilha PV que especifica os vínculos entre os elementos da pilha PS ; e a pilha PSC que contém segmentos coesivamente vinculados.

Uma sentença S será deslocada para a pilha PS se S apresentar um vínculo coesivo com o segmento no topo da pilha ou se S elaborar a mesma entidade que o segmento no topo da pilha (consultando a informação na pilha PV). Se S não satisfizer as restrições anteriores, a pilha PS deverá ser reduzida, utilizando os padrões (5.23) e (5.24) até encontrar no topo da pilha um segmento ao qual S possa ser vinculada ou até que a pilha contenha um único segmento. Se o processo de redução finalizar com um único segmento Seg ao qual S não pode ser vinculada, então Seg será retirado da pilha e passado para a pilha de segmentos já construídos PSC .

A pilha PS pode conter então as seguintes possíveis configurações:

$$\underbrace{\overbrace{\dots, S_0, S_1, \dots, S_k}^{\text{paralelo}}}_{PS} \quad (5.26)$$

$$\underbrace{\overbrace{\dots, S_1, \dots, S_k}^{\text{paralelo}}}_{PS} \quad (5.27)$$

A partir de uma sentença S que elabora uma entidade do segmento S_0 transforma-se a configuração (a) na configuração (b):

$$(a) \underbrace{\overbrace{S_0, S_1, \dots, S_k}^{\text{paralelo}}}_{PS} \Rightarrow (b) \underbrace{\overbrace{S_0, S_1, \dots, S_k, S}^{\text{paralelo}}}_{PS} \quad (5.28)$$

A partir de uma sentença S que elabora uma entidade do segmento S_k transforma-se a configuração (c) na configuração (d):

$$(c) \underbrace{\overbrace{S_1, \dots, S_k}_{PS}}^{\text{paralelo}} \Rightarrow (d) \underbrace{\overbrace{S_1, \dots, S_k, S}^{\text{paralelo}}}_{PS} \quad (5.29)$$

Quando acontece uma mudança de categoria, a pilha de segmentos PS é reduzida até que seja obtido um único segmento Seg que será vinculado aos segmentos contidos na pilha PSC por uma relação de seqüência. Assim, se Seg_1, \dots, Seg_k são os segmentos na pilha PSC , então será construído o segmento Seg para a categoria CA :

$$Seg \leftarrow Seqüência(Seg_1, \dots, Seg_k, Seg).$$

O segmento Seg e a sua categoria $Categoria$ são passados para o processo que vincula Seg aos outros segmentos do sumário. A vinculação é realizada concatenando o novo segmento aos anteriores; assim, se $EstruturaTextual$ contiver a estrutura do sumário até esse momento, o novo segmento será vinculado segundo:

$$EstruturaTextual \leftarrow EstruturaTextual.Categoria(Seg). \quad (5.30)$$

O processo completo de montagem da estrutura textual é implementado pelo predicado Prolog *coerencia*. A análise começa com uma invocação ao predicado Prolog *iniciar_analise* que cria as pilhas e processa a primeira sentença do sumário. O processamento das sentenças que seguem é realizado pelo predicado Prolog *um_passo(I)* encarregado da montagem da sentença de número I . A impressão da estrutura do sumário é realizada pelo predicado Prolog *imprimir_sumario*.

Exemplo 41 Consideremos o texto do sumário da Figura 5.28.

Neste texto as sentenças (1) e (2) pertencem a um segmento de tipo contextual. A sentença (2) está vinculada a sentença (1), pois a entidade "os programas de avaliação" é introduzida em (1) e anaforicamente retomada em (2). A sentença (2) é então introduzida na pilha de segmentos PS que contém a sentença (1). A configuração da pilha é:

(1) Os programas de avaliação de docentes pelos alunos vêm sendo utilizados nos Estados Unidos desde a década de quarenta. (2) Este tipo de programa não é muito comum nas universidades brasileiras. (3) No Departamento de Engenharia Mecânica da UNICAMP foram feitas (...) [Milanez 1987]

Figura 5.28: Vinculação de sentenças (1)

elabora
 $\overbrace{1, 2}$

A sentença (3) é classificada como pertencente a um segmento de tipo experimento. Portanto a pilha deve ser reduzida utilizando o padrão (5.23), para obter o segmento:

Elaboração(1, 2).

Exemplo 42 Consideremos o texto do sumário da Figura 5.29.

(1) O trabalho apresenta a disciplina Similitude em Engenharia. (2) Esta disciplina é ministrada na UFU. (3) O ciclo de aulas visa analisar protótipos. (4) Oito trabalhos foram selecionados (...)

Figura 5.29: Vinculação de sentenças (2)

Neste texto as sentenças (1), (2) e (3) pertencem a um segmento de tipo objetivo. As sentenças (2) e (3) elaboram a sentença (1) já que ambas contém entidades vinculadas à disciplina Similitude em Engenharia. A sentença (4) não pertence ao segmento sendo construído. A configuração da pilha no momento de processar essa sentença é:

elabora
 $\overbrace{\text{paralelo}}$
 $\overbrace{1, 2, 3}$

Portanto a pilha deve ser reduzida utilizando os padrões (5.24) e (5.23), para obter o segmento:

Elaboração(1, Paralelo(2, 3)).

5.4 Conclusões

Neste capítulo tentamos tratar a segunda questão abordada nesta dissertação: **o cálculo da estrutura de um sumário em língua portuguesa**. Apresentamos os mecanismos necessários para a análise de um sumário em língua portuguesa via computador especificando um analisador textual que contém três componentes fundamentais: (i) **o analisador das sentenças**, (ii) **o classificador das sentenças**; e (iii) **o montador da estrutura textual**. O analisador das sentenças produz, a partir de uma sentença em língua portuguesa, entidades, proposições e relações proposicionais que representam o conteúdo proposicional da sentença. O analisador sentencial constrói entidades a partir dos nominais e sua interpretação, proposições a partir dos elementos predicativos da sentença, e relações proposicionais a partir de estruturas sintáticas. O classificador das sentenças é o encarregado da obtenção de uma categoria informacional para uma sentença; a escolha é baseada nos predicadores produzidos pela análise sentencial. Sentenças com a mesma categoria informacional são consideradas como pertencentes ao mesmo segmento informacional do sumário. Finalmente o montador da estrutura textual é o encarregado da construção de segmentos a partir de sentenças. Os segmentos de texto são construídos utilizando-se relações coesivas entre sentenças. Para testar o modelo proposto foi implementado um protótipo experimental na linguagem de programação Prolog. O analisador sentencial foi obtido a partir de regras DCGs, utilizando-se conceitos de Slot Grammars para a análise dos predicativos. O protótipo desenvolvido constrói a partir de um sumário em língua portuguesa a representação textual definida nesta dissertação.

Capítulo 6

Conclusões

Nesta dissertação abordamos o problema da análise de um texto em língua portuguesa, com ênfase no tratamento de fenômenos essenciais para a obtenção de uma tradução automática de alta qualidade. Dada a complexidade da análise de um texto irrestrito, escolhemos trabalhar com sumários de artigos técnicos por duas razões: em primeiro lugar um sumário, embora sendo curto, é um texto e como tal apresenta os fenômenos interessantes de coesão e coerência de outros textos. Em segundo lugar, existe um interesse prático na produção de sumários em outras línguas, sendo então um domínio apropriado para o estudo dos problemas relacionados à tradução automática.

Os sistemas de tradução podem ser projetados considerando duas etapas independentes: por um lado o processo de análise do texto na língua origem, e por outro, o processo de geração do texto na língua destino. A arquitetura transfer torna possível a especificação de um analisador ainda desconhecendo a língua para a qual será feita a tradução. Por isso, nesta dissertação descrevemos uma etapa de análise de sumários em língua portuguesa, que poderá ser incorporada a um sistema de tradução de sumários para outras línguas.

Fazendo uma análise do tratamento de fenômenos lingüísticos em sistemas de tradução automática observamos que em geral a ênfase está no tratamento da sentença como unidade fundamental. Entretanto, na tradução humana o texto como unidade e a sua estrutura são levados em consideração na hora de traduzir. Baseando-nos neste fato decidimos abordar o problema da estrutura de um sumário como meio para melhorar o processo de tradução automática. Para isso nesta dissertação duas questões foram abordadas: **a estrutura dos sumários e o cálculo dessa estrutura.**

Na Figura 6.1 resumizamos o trabalho desenvolvido nesta pesquisa. A partir da análise de um corpus de sumários reais definimos uma representação para a estrutura dos sumários e um processo computacional que realiza o mapeamento entre um sumário e a sua estrutura.

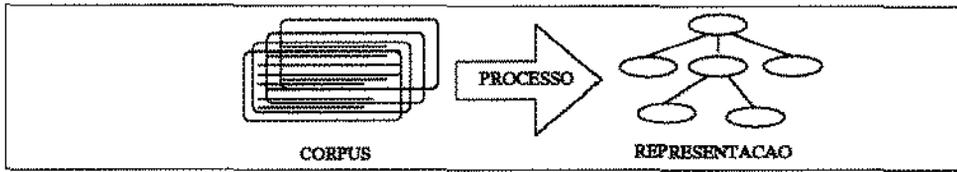


Figura 6.1: O trabalho desenvolvido

A estrutura definida nesta dissertação consta de três componentes: **o conteúdo proposicional do sumário**, **o conteúdo informacional** e **as relações entre as proposições do sumário**. A estrutura pode ser representada graficamente por uma árvore. O processo de cálculo especificado leva em consideração a **análise a nível sentencial** e a **análise a nível textual**, sendo a coesão referencial o principal fenômeno intersentencial abordado nesta dissertação. A decisão do tratamento da coesão referencial como principal fenômeno textual foi baseada na análise dos textos do corpus e na verificação da utilização de anáfora definida como o principal mecanismo de vinculação entre sentenças.

Podemos sumarizar as contribuições deste estudo em três pontos:

- (i) definimos uma representação para a estrutura de um sumário;
- (ii) especificamos um processo de análise textual para sumários; e
- (iii) abordamos o problema da coesão referencial para a língua portuguesa em um sistema automático.

Como parte desta pesquisa implementamos na linguagem de programação Prolog um protótipo do analisador textual especificado nesta dissertação. Para o analisador a nível sentencial utilizamos gramáticas de cláusulas definidas (DCGs) acrescidas de mecanismos utilizados em Slot Grammars. Para o cálculo da estrutura textual baseamo-nos na resolução de referências anafóricas. Para o particionamento do texto utilizamos o conhecimento sobre os predicadores marcadores de segmentos informacionais.

Nesta dissertação não foi abordado o problema da ambigüidade existente em todos os níveis da análise textual: lexical, sintática, semântica e referencial. Portanto, um estudo sobre como lidar com esse problema deve ser iniciado.

As sentenças e frases tratadas pelo analisador sentencial implementado apenas refletem os padrões de produção textual observados no corpus; não pretendemos propor uma teoria geral sobre a análise sintática da língua portuguesa por computador; ao contrário, utilizamos o analisador como meio para a obtenção dos elementos necessários para a construção da representação proposta nesta dissertação. Entretanto, acreditamos que a abordagem adotada para o tratamento

de sentenças em língua portuguesa possa ser enriquecida para abranger um conjunto maior de fenômenos lingüísticos.

Para a determinação dos segmentos textuais, utilizamos predicativos que surgiram como indicadores de segmentos informacionais, a partir do estudo do corpus de sumários e das informações teóricas obtidas. Entretanto, acreditamos que outros mecanismos possam ser utilizados para o particionamento, tais como marcadores discursivos. Na nossa abordagem os segmentos informacionais não possuem sub-segmentos. Esta hipótese, embora válida para os sumários analisados, não é válida para textos em geral; portanto deve ser revisada se pretendermos aplicar o modelo a outros tipos de texto.

O cálculo do conteúdo proposicional do texto é baseado na estrutura sintática dos sintagmas nominais e na estrutura argumental dos predicativos. Existem, entretanto, relações de natureza mais profunda que não foram abordadas nesta dissertação, tais como as relações entre um núcleo nominal e os seus complementos. Reconhecemos uma limitação nas relações proposicionais adotadas. Elas foram por nós determinadas a partir de construções sintáticas que marcam relações lógico-semânticas e a partir de vínculos coesivos observados no corpus. A presença de uma construção sintática sem dúvida marca uma relação semântica; entretanto, para a verificação de relações mais profundas serão necessários conhecimentos adicionais, tais como os apontados no Capítulo 3 desta dissertação.

Nesta dissertação não abordamos outras formas de coesão referencial para o português, além de anáfora definida. O tratamento da coesão seqüencial, que inclui a seqüenciação frásica e a estrutura temática, também não foi abordado; entretanto, reconhecemos que são importantes para o processamento textual e em particular devem ser levados em consideração em tradução automática.

A representação definida nesta dissertação surgiu a partir do estudo de um subconjunto de textos reais; portanto uma das primeiras questões a serem consideradas é se a representação pode ser estendida para tratar sumários em geral e, mais ainda, se ela poderia ser estendida para o tratamento de texto irrestrito (veja Figura 6.2). A representação por nós definida contém os traços interessantes de uma representação textual: segmentos, proposições e relações proposicionais; por isso acreditamos que possa ser utilizada em forma quase direta para outros domínios textuais.

As limitações deste trabalho sugerem algumas propostas de trabalhos futuros nesta área de processamento de textos por computador. Dentre esses estudos podemos considerar:

- (i) tratamento computacional de outras formas de coesão referencial para a língua portuguesa, tais como as apresentadas no Capítulo 3;

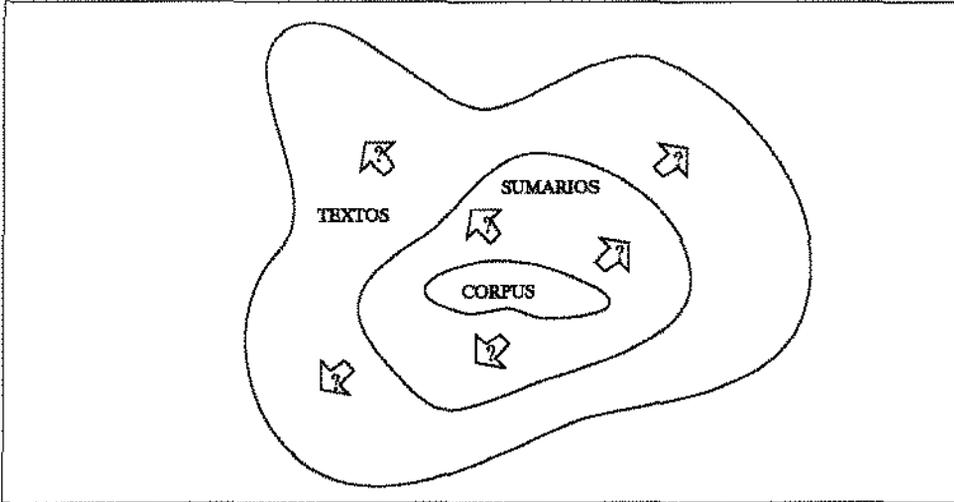


Figura 6.2: Extensões

- (ii) tratamento da coesão seqüencial como outro caminho para o cálculo da estrutura de um texto;
- (iii) abordagem computacional ao problema da coerência textual incorporando conhecimento do mundo;
- (iv) teste da representação proposta para gerar um sumário na língua portuguesa ou em outras línguas, completando desta forma o processo de tradução automática de sumários. Para isso um sistema de geração de sumários deverá ser especificado, assim como um processo de conversão entre representações.
- (v) enriquecimento da representação proposta, ampliando por um lado o conjunto de relações proposicionais, e por outro os segmentos informacionais.

Apêndice A

Exemplos de análise textual

A.1 Exemplo I

os trocadores de calor sao elementos basicos.
o trabalho apresenta um metodo.
o metodo e usado para o teste dos trocadores de calor.
usando o metodo recomendado os alunos podem estimar a queda da pressao.

-

```
| ?- dicionario.  
Incorporando pronomes...  
Incorporando determinantes...  
Incorporando nomes comuns...  
Incorporando nomes proprios...  
Incorporando predicativos...  
Incorporando verbos...  
Incorporando adjetivos...  
Incorporando conectivos...  
Incorporando adverbios...  
Incorporando preposicoes...  
Incorporando contracoes...  
Incorporando pontuacao...  
Incorporando conjuncao...  
Incorporando coordenacao...  
Incorporando relativos...
```

```
| ?- parsear_sumario(trocadores).  
oracao 1 =>[os,trocadores de calor,sao,elementos,basicos]  
Parseada
```

```

oracao 2 =>[o, trabalho, apresenta, um, metodo]
Parseada
oracao 3 =>[o, metodo, e usado, para, o, teste, dos, trocadores de calor]
Parseada
oracao 4 =>[usando, o, metodo, recomendado, os, alunos, podem,
estimar, a, queda da pressao]
Parseada

```

```

| ?- coerencia.
Coerencia das sentencas...
Iniciando montagem textual...
Passos...
Um passo...2
Um passo...3
Um passo...4
Fim da montagem...
Construindo estrutura...

```

```

| ?- imprimir_sumario.
COMEÇO SUMARIO
CONTEXTO
ser(e9,e18)

```

```

OBJETIVO

```

```

Elaboracao
apresentar(e6,x1)
usar(x1, testar(e9))

```

```

CONCLUSAO

```

```

Ocasiao

```

```

Elaboracao
usar(x2)
recomendar(x2)
poder(e10, estimar(e17))

```

```

FIM SUMARIO

```

A.2 Exemplo II

com a utilizacao de calculador torna-se necessaria a adequacao do plano de ensino de Calculo Numerico para a formacao dos engenheiros.

algumas recomendacoes sobre a utilizacao de calculador sao feitas.

algumas alteracoes na metodologia de ensino desta disciplina sao sugeridas.

-

```
| ?- parsear_sumario(teste1).
```

```
oracao 1 =>[com,a,utilizacao,de,calculador,torna-se,necessaria,
a,adequacao,do,plano de ensino,de,Calculo
Numerico,para,a,formacao,dos,engenheiros]
```

```
Parseada
```

```
oracao 2 =>[algumas,recomendacoes,sobre,
a,utilizacao,de,calculador,sao_feitas]
```

```
Parseada
```

```
oracao 3 =>[algumas,alteracoes,na,metodologia
de ensino,desta,disciplina,sao_sugeridas]
```

```
Parseada
```

```
| ?- coerencia.
```

```
Coerencia das sentencas...
```

```
Iniciando montagem textual...
```

```
Passos...
```

```
Um passo...2
```

```
Um passo...3
```

```
Fim da montagem...
```

```
Construindo estrutura...
```

```
| ?- imprimir_sumario.
```

```
COMEÇO SUMARIO
```

```
CONTEXTO
```

```
Causa/Consequencia
```

```
utilizar(e15)
```

```
necessario(adequar(p(e1),formar(e16)))
```

CONCLUSAO

Sequencia

sugerir(alterar(m(e1)))

fazer(recomendar(utilizar(e15)))

FIM SUMARIO

A.3 Exemplo III

o trabalho tem como meta divulgar a disciplina
Similitude em Engenharia ministrada no curso
de engenharia mecanica da UFU.

esta disciplina visa proporcionar aos alunos
fundamentos da teoria de modelos.

o ciclo de aulas tem como objetivos distinguir
as quantidades envolvidas em um experimento
coma manipular dados e analisar prototipos
atraves de modelos.

oito trabalhos de laboratorio envolvendo
a area Mecanica dos Fluidos foram selecionados.

| ?- parsear_sumario(teste2).

oracao 1 =>[o, trabalho, tem-como-meta, divulgar,
a, disciplina, Similitude em Engenharia, ministrada,
no, curso, de, engenharia mecanica, da, UFU]

Parseada

oracao 2 =>[esta, disciplina, visa, proporcionar,
Aos, alunos, fundamentos, da, teoria de modelos]

Parseada

oracao 3 =>[o, ciclo, de, aulas, tem-como-objetivos,
distinguir, as, quantidades, envolvidas, em, um, experimento,
coma, manipular, dados, E, analisar, prototipos, atraves de, modelos]

Parseada

oracao 4 =>[oito, trabalhos de laboratorio, envolvendo,
a, area, Mecanica dos Fluidos, foram-selecionados]

Parseada

| ?- coerencia.

Coerencia das sentencas...
 Iniciando montagem textual...
 Passos...
 Um passo...2
 Um passo...3
 Um passo...4
 Fim da montagem...
 Construindo estrutura...

| ?- imprimir_sumario.
 COMECO SUMARIO
 OBJETIVO

Elaboracao
 objetivo(e6,elaboracao(divulgar(e4),ministrar(e4,c(e7))))
 objetivo(e4,proporcionar(a(e4),f(e8)))

EXPERIMENTO
 objetivo(c(e4),sequencia(elaboracao(distinguir(x3),envolver(x3,x1)),
 manipular(e11),analisar(e12,e13)))

METODOLOGIA

Elaboracao
 envolver(x4,e9)
 selecao(x4)

FIM SUMARIO

A.4 Exemplo IV

a engenharia depende das ciencias mas
 a engenharia depende da criatividade.
 um experimento foi_realizado para o desenvolvimento
 da criatividade na engenharia.o trabalho apresenta
 as experiencias realizadas com estudantes
 para o desenvolvimento da criatividade na engenharia.

-

```

| ?- parsear_sumario(teste3).
oracao 1 =>[a,engenharia,depende,das,ciencias,mas,
a,engenharia,depende,da,criatividade]
Parseada
oracao 2 =>[um,experimento,foi_realizado,
para,o,desenvolvimento,da,criatividade,na,engenharia]
Parseada
oracao 3 =>[o,trabalho,apresenta,as,experiencias,realizadas,
com,estudantes,para,o,desenvolvimento,
da,criatividade,na,engenharia]
Parseada

| ?- coerencia.
Coerencia das sentencas...
Iniciando montagem textual...
Passos...
Um passo...2
Um passo...3
Fim da montagem...
Construindo estrutura...

| ?- imprimir_sumario.
COMEÇO SUMARIO
CONTEXTO
oposicao(depender(e2,e3),depender(e2,e20))

EXPERIMENTO
realizar(x1,desenvolver(e20,e2))

OBJETIVO

Elaboracao
apresentar(e6,x2)
realizar(x2,e19,desenvolver(e20,e2))

FIM SUMARIO

```

Apêndice B

Código do analisador

```
% dicionario
```

```
% incorporar pronomes
```

```
dicionario:-  
write('Incorporando pronomes...'),nl,  
pronome(X1,X2,X3),  
assert(dicionario(X1,pronome(X1,X2,X3))),  
retract(pronome(X1,X2,X3)),fail.
```

```
% incorporar determinantes ao dicionario
```

```
dicionario:-  
write('Incorporando determinantes...'),nl,  
determinante(X1,X2,X3,X4),  
assert(dicionario(X1,determinante(X1,X2,X3,X4))),  
retract(determinante(X1,X2,X3,X4)),fail.
```

```
% incorporar nomes
```

```
dicionario:-  
write('Incorporando nomes comuns...'),nl,  
nome(X1,X2,X3,X4,X5,X6,X7),  
assert(dicionario(X1,nome(X1,X2,X3,X4,X5,X6,X7))),  
retract(nome(X1,X2,X3,X4,X5,X6,X7)),fail.
```

```
% incorporar nomes proprios
```

```

diccionario:-
write('Incorporando nomes propios...'),nl,
nome_prop(X1,X2,X3,X4,X5,X6,X7,X8,X9),
assert(diccionario(X1,nome_prop(X1,X2,X3,X4,X5,X6,X7,X8,X9))),
retract(nome_prop(X1,X2,X3,X4,X5,X6,X7,X8,X9)),fail.

```

```
% incorporar predicativos
```

```

diccionario:-
write('Incorporando predicativos...'),nl,
predicativo(X1,X2,X3,X4,X5,X6,X7),
assert(diccionario(X1,predicativo(X1,X2,X3,X4,X5,X6,X7))),
retract(predicativo(X1,X2,X3,X4,X5,X6,X7)),fail.

```

```
% incorporar verbos
```

```

diccionario:-
write('Incorporando verbos...'),nl,
verbo(X1,X2,X3,X5,X6,X7,X8,X9,X10),
assert(diccionario(X1,verbo(X1,X2,X3,X5,X6,X7,X8,X9,X10))),
retract(verbo(X1,X2,X3,X5,X6,X7,X8,X9,X10)),fail.

```

```
% incorporar adjetivos
```

```

diccionario:-
write('Incorporando adjetivos...'),nl,
a(X1,X2,X5,X6),
assert(diccionario(X1,a(X1,X2,X5,X6))),
retract(a(X1,X2,X5,X6)),fail.

```

```
% incorporar conectivos
```

```

diccionario:-
write('Incorporando conectivos...'),nl,
conectivo(X1,X2,X3,X4),
assert(diccionario(X1,conectivo(X1,X2,X3,X4))),
retract(conectivo(X1,X2,X3,X4)),fail.

```

```
% incorporar adverbios
```

```

diccionario:-
write('Incorporando adverbios...'),nl,
adv(X1,X2,X3,X4,X5),
assert(diccionario(X1,adv(X1,X2,X3,X4,X5))),

```

```
retract(adv(X1,X2,X3,X4,X5)),fail.
```

```
% incorporar preposicoes
```

```
dicionario:-  
write('Incorporando preposicoes...'),nl,  
prep(X1,X2,X3),  
assert(dicionario(X1,prep(X1,X2,X3))),  
retract(prep(X1,X2,X3)),fail.
```

```
% incorporar contracoes
```

```
dicionario:-  
write('Incorporando contracoes...'),nl,  
contracao(X1,X2,X3),  
assert(dicionario(X1,contracao(X1,X2,X3))),  
retract(contracao(X1,X2,X3)),fail.
```

```
% incorporar comas
```

```
dicionario:-  
write('Incorporando pontuacao...'),nl,  
coma(X1),  
assert(dicionario(X1,coma(X1))),  
retract(coma(X1)),fail.
```

```
% incorporar coordenacao
```

```
dicionario:-  
write('Incorporando conjuncao...'),nl,  
conjuncao(X1),  
assert(dicionario(X1,conjuncao(X1))),  
retract(conjuncao(X1)),fail.
```

```
% coordenacao
```

```
dicionario:-  
write('Incorporando coordenacao...'),nl,  
coordenacao(X1,X2),  
assert(dicionario(X1,coordenacao(X1,X2))),  
retract(coordenacao(X1,X2)),fail.
```

```
% relativos
```

```

dicionario:-
write('Incorporando relativos...'),nl,
relativo(X1),
assert(dicionario(X1,relativo(X1))),
retract(relativo(X1)),fail.

% fim da incorporacao

dicionario:-iniciar_evento,iniciar_entidade,iniciar_pronome.

% DETERMINANTES

determinante(o,det,mas,sin).
.
.
.

% PRONOMES

pronome(ele,mas,sin).
.
.
.

% NOMES COMUNS

nome(disciplina,disciplina,[],incompleto,disciplina,fem,sin).
.
.
.

% NOMES PROPRIOS

nome_prop('Calculo Numerico',
disciplina,[],completo,X,disciplina(X,nome=calculo_numerico,det,fem,sin),
nome=calculo_numerico,mas,sin).
.
.
.

% PREPOSICOES

```

```
prep(de,p(de)).
```

```
.  
. .  
. .
```

```
% VERBOS
```

```
verbo(envolvendo,envolver,[[paciente,np],[fonte,np]],  
envolver,ger,_,pre,ger,pro).
```

```
.  
. .  
. .
```

```
% CONTRACOES
```

```
contracao(do,de,o).
```

```
.  
. .  
. .
```

```
% ADJETIVOS
```

```
a(simples,simplicidade,qualificador=simples,_,_).
```

```
% PREDICATIVOS
```

```
predicativo(necessaria,necessidade,[[objeto,np]],  
X,necessario(X),fem,sin).
```

```
.  
. .  
. .
```

```
% COMA
```

```
coma(coma).
```

```
% conjuncao
```

```
conjuncao('E').
```

```
% COORDENACA
```

```
coordenacao(mas,oposicao).
```

```
% RELATIVOS
```

```
relativo(que).
```

```
% coordenacao
```

```
coordenacao(coordenacao(C),Sem,Var) -->
{instanciar_evento(Var)},
[coordenacao(C,Sem)].
```

```
% relativo
```

```
rel(rel(R)) -->
[relativo(R)].
```

```
% busca em dicionario
```

```
look-up([],[]).
```

```
look-up([Palavra|Palavras],[Termo1,Termo2|Termos]):-
dicionario(Palavra,Termo),
Termo=..[contracao,Palavra,Preposicao,Artigo],
dicionario(Preposicao,Termo1),
dicionario(Artigo,Termo2),
look-up(Palavras,Termos).
```

```
look-up([Palavra|Palavras],[Termo|Termos]):-
dicionario(Palavra,Termo),
look-up(Palavras,Termos).
```

```
% parsing do sumario
```

```
parsear_sumario(Sumario):-
apagar_oracoes,
apagar_arvores,
apagar_tokens,
apagar_verbos,
apagar_descricoes,
apagar_tempo,
```

```

apagar_aspecto,
apagar_auxiliares,
apagar_auxiliares2,
apagar_ativados,
apagar_arvore2,
procesar(Sumario),
parsear_oracao(1).

```

```

apagar_auxiliares:-
apagar_sub_segmento,
apagar_segmento2.

```

```

apagar_segmento2:-
retract(segmento(I,S)),fail.

```

```

apagar_segmento2.

```

```

apagar_sub_segmento:-
retract(sub_segmento(S)),fail.

```

```

apagar_sub_segmento.

```

```

coerencia:-
write('Coerencia das sentencas...'),nl,
coerencia(1),
referentes,
write('Iniciando montagem textual...'),nl,
iniciar_analisis,
write('Passos...'),nl,
passos(2),
write('Fim da montagem...'),nl,
finalizar_analisis1,
construir_arvores(1),
write('Construindo estrutura...'),nl,
construir_sumario.

```

```

coerencia(I):-
coerencia1(I),
J is I+1,
coerencia(J).

```

```
coerencia(_).
```

```
passos(I):-
segmento(I,S),
write('Um passo...'),write(I),nl,
um_passo(I),
J is I+1,
passos(J).
```

```
passos(_).
```

```
construir_arvores(I):-
construir_arvore(I,A),
J is I+1,
construir_arvores(J).
```

```
construir_arvores(_).
```

```
% processador da coerencia textual
```

```
coerencial(I):-
verbos(I,V),
descricaoes(I,D),
merge(V,D,T),
processar_termos(T,Res),
assert(segmento(I,Res)),
criar_segmentos(Res, []).
```

```
criar_segmentos([],Resultado):-assert(sub_segmento(Resultado)).
```

```
criar_segmentos([Predicado|Predicados],Principio):-
Predicado=.. [Functor|Args],
sintatico(Functor),
!,
quantidade(Args,N),
extraer(Predicados,N,Predicados1,Predicados2),
assert(sub_segmento(Predicados1)),
name(Functor,Nome),
name(Functor1,[120|Nome]),
Predicado1=.. [Functor1|Args],
merge(Principio,[Predicado1|Predicados2],Resultado),
criar_segmentos(Resultado,[]).
```

```

criar_segmentos([Predicado|Predicados],Principio):-
merge(Principio,[Predicado],Resultado),
criar_segmentos(Predicados,Resultado).

quantidade([],0).

quantidade([A|B],N):-quantidade(B,M),N is M + 1.

extraer(Predicados,0,[],Predicados).

extraer([Predicado|Predicados],N,[Predicado|Resto],Resultado):-
N > 0,
M is N - 1,
extraer(Predicados,M,Resto,Resultado).

sintatico(sequencia).

% processar os termos das oracoes

processar_termos([],[]).

processar_termos([T|Termos],[Resultado|Resultados]):-
processar_termo(T,Resultado),
processar_termos(Termos,Resultados).

processar_termo(Termo,Resultado):-
Termo=..[Cabeca|Calda],
processar_parametros(Calda,Variaveis),
Resultado=..[Cabeca|Variaveis].

processar_parametros([],[]).

processar_parametros([Parametro|Parametros],Variaveis):-
processar_parametro(Parametro,Par),
processar_parametros(Parametros,Pars),
merge(Par,Pars,Variaveis).

processar_parametro(tempo=X,[]):-!.

processar_parametro(Rol=indefinido,[]):-!.

processar_parametro(Rol=Variavel,[Variavel]):-!.

```

```

processar_parametro(fem, []):-!.
processar_parametro(mas, []):-!.
processar_parametro(sin, []):-!.
processar_parametro(plu, []):-!.
processar_parametro(none, []):-!.
processar_parametro(det, []):-!.
processar_parametro(nodet, []):-!.
processar_parametro(dem1, []):-!.
processar_parametro(dem2, []):-!.
processar_parametro(dem3, []):-!.
processar_parametro(8, []):-!.
processar_parametro(algum, []):-!.
processar_parametro(Variavel, [Variavel]):-!.

% eliminar processamento previo

apagar_oracoes:-retract(oracao(I,S)),fail.

apagar_oracoes.

apagar_arvores:-retract(arvore(I,S)),fail.

apagar_arvores.

apagar_tokens:-retract(tokens(I,S)),fail.

apagar_tokens.

apagar_verbos:-retract(verbos(I,V)),fail.

apagar_verbos.

apagar_descricoes:-retract(descricoes(I,D)),fail.

apagar_descricoes.

apagar_tempo:-retract(tempo(I,T)),fail.

apagar_tempo.

apagar_aspecto:-retract(aspecto(I,A)),fail.

apagar_aspecto.

```

```
% processar sumario completo
```

```
procesar(Sumario):-
  iniciar_evento,iniciar_entidade,iniciar_pa,iniciar_la,
  see(Sumario),
  leer_sentencas(1),
  seen.
```

```
% leer cada senntenca do sumario
```

```
leer_sentencas(I):-read_sent(S),
  (S=[];assert(oracao(I,S)),J is I+1,leer_sentencas(J)).
```

```
% parsing das oracoes
```

```
parsear_oracao(I):-
  oracao(I,S),
  write(oracao),write(' '),
  write(I),write(' '),
  write('=>'),
  write(S),nl,
  look-up(S,S1),assert(tokens(I,S1)),
  (parsear1(S1,Arvore,Lista1,Lista2,Tempo,Aspecto),
  write('Parseada'),nl,
  assert(arvore(I,Arvore)),
  assert(descricoes(I,Lista1)),
  assert(verbos(I,Lista2)),
  assert(tempo(I,Tempo)),
  assert(aspecto(I,Aspecto));true,write('nao parseada'),nl),
  J is I+1,
  parsear_oracao(J).
```

```
parsear_oracao(_).
```

```
% parsing das sentencas
```

```
parsear1(L,Arvore,Lista1,Lista2,Tempo,Aspecto):-
  s(Tipo,Arvore,S_var,_,Lista1,Lista2,Tempo,Aspecto,L,[]).
```

```
% sentenca simples
```

```
s(fin,s(NP_arvore,VP_arvore),Var,_,Entidades,Predicados,Tempo,Aspecto) -->
```

```

    np(X,Entidade,NP_arvore,Gen,Num,Entidades1,Predicados1,arg),
    vp(fin,[[np=X]],VP_arvore,Var,Num,Tempo,Aspecto,Entidades2,Predicados2),
    {merge(Entidades1,Entidades2,Entidades),
    merge(Predicados1,Predicados2,Predicados)}.

```

```

s(fin,s(np(nil),VP_arvore),Var,_,Entidades,Predicados,Tempo,Aspecto) -->
vp(fin,[[s(ger)=x]],VP_arvore,Var,Num,Tempo,Aspecto,
Entidades,Predicados).

```

```

s(fin,s(np(nil),VP_arvore),Var,_,Entidades,Predicados,Tempo,Aspecto) -->
vp(fin,[[np=indefinido]],VP_arvore,Var,Num,Tempo,Aspecto,
Entidades,Predicados).

```

% sentença causativa

```

s(fin,Arvore,Var,_,Entidades,Predicados,Tempo,Aspecto) -->
pp(X,Entidade,PP_arvore,Entidades1,Predicados1,arg),
vp(fin,[[pp(P)=X]],VP_arvore,Var,_,Tempo,Aspecto,
Entidades2,Predicados2),
{merge(Entidades1,Entidades2,Entidades),
merge(Predicados1,Predicados2,Predicados)}.

```

% oracao reduzida com verbo no gerundio (sem sujeito)

```

s(ger,s(VP_arvore),Var,X,Num,Entidades,Predicados) -->
vp(ger,[X],VP_arvore,Var,Num,Tempo,Aspecto,Entidades,Predicados).

```

% oracao reduzida com verbo no infinitivo (sem sujeito)

```

s(infinitivo,Arvores,Var,X,Entidades,Predicados) -->
s(inf,Arvores,Var,X,Entidades,Predicados).

```

```

s(infinitivo,Arvores,Var,X,Entidades,Predicados) -->
s(con,Arvores,Var,X,Entidades,Predicados).

```

```

s(inf,s(np(nil),Arvores),Var,X,Entidades,Predicados) -->
vp(inf,[],Arvores,Var,Num,Tempo,Aspecto,Entidades,Predicados).

```

```

s(con,Arvore,Var,X,Entidades,Predicados) -->
s(inf,Arvore1,Var1,X,Entidades1,Predicados1),
resto(Arvores2,Vars,X,Entidades2,Predicados2),
{Arvore=.. [conjuncao,Arvore1|Arvores2]},
instanciar_evento(Var),
Conj=.. [sequencia,Var,Var1|Vars].

```

```
merge(Entidades1,Entidades2,Entidades),
merge([Conj|Predicados1],Predicados2,Predicados)}.
```

```
resto([Arvore|Arvores],[Var|Vars],X,Entidades,Predicados,[coma(COMA)|T],T1):-
s(inf,Arvore,Var,X,Entidades1,Predicados1,T,T2),
resto(Arvores,Vars,X,Entidades2,Predicados2,T2,T1),
merge(Entidades1,Entidades2,Entidades),
merge(Predicados1,Predicados2,Predicados).
```

```
resto([Arvore],[Var],X,Entidades,Predicados,[conjuncao(E)|T],T1):-
s(inf,Arvore,Var,X,Entidades,Predicados,T,T1).
```

```
% oracao com participio, modificadora de um NP
```

```
s(pred,S_arvore,Var,X,Num,Entidades,[Pred|Predicados]) -->
pred(Slots,Var,Sem,P_arvore,Gen,Num,Ent1,Prop1),
fill1([X],Slots,Roles,Arvores,Entidades,Predicados),
{S_arvore=..[s,P_arvore|Arvores],
Sem=..[F,Var],
Pred=..[F,Var|Roles],
instanciar_evento(Var)}.
```

```
% oracao adjetiva, complemento de um verbo
```

```
s(pred,S_arvore,Var,X,Entidades,[Pred|Predicados]) -->
pred(Slots,Var,Sem,P_arvore,Gen,Num,Ent1,Prop1),
fill1([],Slots,Roles,Arvores,Entidades,Predicados),
{S_arvore=..[s,P_arvore|Arvores],
Sem=..[F,Var],
Pred=..[F,Var|Roles],
instanciar_evento(Var)}.
```

```
% elemento predicativo(=participio)
```

```
pred(Slots,Var,Sem,pred(Var,P),Gen,Num,[],[Var]) -->
[predicativo(P,Tipo,Slots,Var,Sem,Gen,Num)].
```

```
% sentenca coordenada de duas sentencas
```

```
s(coor,s(S_arvore1,Coord,S_arvore2),Var,_,Entidades,Predicados,
Tempo,Aspecto) -->
```

```

s(fin,S_arvore1,Var1,_,Entidades1,Predicados1,Tempo1,Aspecto1),
coordenacao(Coord,Pred,Var),
s(fin,S_arvore2,Var2,_,Entidades2,Predicados2,Tempo2,Aspecto2),
{merge(Entidades1,Entidades2,Entidades),
Sem=..[Pred,Var,Var1,Var2],
merge(Predicados1,[Sem|Predicados2],Predicados)}.

```

% sentença com oração reduzida de gerundio

```

s(sub,s(S_arvore1,S_arvore2),Var,_,Entidades,Predicados,
Tempo,Aspecto) -->
s(ger,S_arvore1,Var1,[np=indefinido],_,Entidades1,Predicados1),
s(fin,S_arvore2,Var2,_,Entidades2,Predicados2,Tempo,Aspecto),
{merge(Entidades1,Entidades2,Entidades),
merge(Predicados1,[Sem|Predicados2],Predicados),
instanciar_evento(Var),
Sem=..[ocasio,Var,Var1,Var2]}.

```

% sentença relativa complemento de um verbo relativo+sentença

```

s(relativa,s(A_relativo,S_arvore),Var,_,Entidades,Predicados) -->
rel(A_relativo),
s(fin,S_arvore,Var,_,Entidades,Predicados,Tempo,Aspecto).

```

% sintagma verbal

```

vp(Fin,[[Tipo=X]],VP_arvore,VP_var,Num,Tempo,Aspecto,Entidades,
[F1|Predicados]) -->
{Fin==fin;Fin==ger},
v(Slots,F,Verbo,Fin,VP_var,Num,Ent1,Prop1,Tempo,Modo,Aspecto),
fill1([[Tipo=X]],Slots,Roles,Arvores,Entidades1,Predicados1),
completar1(Preds2,Roles1,Arvores1,Ent3,Prop3,acao,
Entidades2,Predicados2),
{merge(Roles,Roles1,Roles2),
F1=..[F,VP_var,tempo=Tempo|Roles2],
merge(Arvores,Arvores1,Arvores0),
VP_arvore=..[vp,Verbo|Arvores0],
merge(Entidades1,Entidades2,Entidades),
merge(Predicados1,Predicados2,Predicados),
instanciar_evento(VP_var)}.

```

```

vp(inf,[],VP_arvore,VP_var,Num,Tempo,Aspecto,Entidades1,
[F1|Predicados1]) -->

```

```
v(Slots,F,Verbo,inf,VP_var,Num,Ent1,Prop1,Tempo,Modo,Aspecto),
fill1([],Slots,Roles,Arvores,Entidades1,Predicados1),
{F1=..[F,VP_var,tempo=Tempo|Roles],
VP_arvore=..[vp,Verbo|Arvores],
instanciar_evento(VP_var)}.
```

```
% verbo
```

```
v(Slots,F,v(Var,V),Finitude,Var,Num,[],[Var],Tempo,Modo,Aspecto) -->
[verbo(V,Tipo,Slots,F,Finitude,Num,Tempo,Modo,Aspecto)].
```

```
% complementos subcategorizados do verbo
```

```
fill1([],Slots,SlotFiller,Aslots,Entidades,Predicados) -->
fill(Slots,SlotFiller,Aslots,_,Entidades,Predicados).
```

```
fill1([A|L],Slots,[Rol|Roles],Aslots,Entidades,Predicados,L1,L2):-
select(Slot,Slots,Slots1),
satisfacer(Slot,A,Rol),
!,
fill1(L,Slots1,Roles,Aslots,Entidades,Predicados,L1,L2).
```

```
satisfacer([TipoSlot,TipoFrase],[TipoFrase=X],TipoSlot=X).
```

```
% sintagma preposicionado
```

```
pp(F,Entidade,pp(Parvore,Narvore),Entidades,Predicados,Arg,[prep(X,S,Parvore)|L]
np(F,Entidade,Narvore,Gen,Num,Entidades,Predicados,Arg,L,L1).
```

```
% tipos de sintagmas nominais
```

```
np(X,Entidade,NP_arvore,Gen,Num,Entidades,[Entidade|Predicados],Arg) -->
det(Q,Adet,Gen,Num),
n(Slots,verbo_nominal,X,Sem,N_arvore,Gen,Num,
Completude,Ent1,Pred1),
fill(Slots,Roles,Arvores,Num,Entidades,Predicados),
{instanciar_evento(X),
merge(Roles,[Q,Gen,Num],Roles1),
Entidade=..[Sem,X|Roles1],
NP_arvore=..[np,Adet,N_arvore|Arvores],
```

atualizar_pa(X, Gen, Num)}.

```
np(Ref, Entidade, np(pronome(Pro, X)), Gen, Num, [Entidade], [], Arg) -->
[pronome(Pro, Gen, Num)],
{instanciar_pronome(X),
Entidade=.. [anafora, X, Gen, Num, Ref],
busca_ref(Gen, Num, Ref)}.
```

```
np(X, Entidade, NP_arvore, Gen, Num, [Entidade|Entidades], Predicados, Arg) -->
det(Q, Adet, Gen, Num),
n(Slots, Tipo, X, Sem, Arvore, Gen, Num, Completude, Ent, Prop),
nproprio(Slots1, Tipo1, Y, Sem1, Sem2, Arvore1, Gen1, Num, Completude1, _),
s(complemento, C_arvore, S_var, [np=X], Num, Entidades, Predicados),
{Entidade=.. [Sem, X, Sem2, Q, Gen, Num],
N_arvore=.. [np, Adet, Arvore, Arvore1]},
interpretar(Entidade, completo),
atualizar_pa(X, Gen, Num)}.
```

```
np(X, Entidade, NP_arvore, Gen, Num, [Entidade|Entidades], Predicados, Arg) -->
det(Q, Adet, Gen, Num),
n(Slots, Tipo, X, Sem, Arvore, Gen, Num, Completude, Ent, Prop),
nproprio(Slots1, Tipo1, Y, Sem1, Sem2, Arvore1, Gen1, Num, Completude1, _),
{Entidade=.. [Sem, X, Sem2, Q, Gen, Num],
N_arvore=.. [np, Adet, Arvore, Arvore1]},
interpretar(Entidade, completo),
atualizar_pa(X, Gen, Num)}.
```

```
s(complemento, Arvore, Var, Adelanto, Num, Entidades, Predicados) -->
s(pred, Arvore, Var, Adelanto, Num, Entidades, Predicados).
```

```
s(complemento, Arvore, Var, Adelanto, Num, Entidades, Predicados) -->
s(ger, Arvore, Var, Adelanto, Num, Entidades, Predicados).
```

```
np(X, Sem, np(Adet, Arvore1), Gen, Num, [Sem], [], Arg) -->
det(Q, Adet, Gen, Num),
nproprio(Slots1, Tipo, X, Sem1, Sem2, Arvore1, Gen, Num, Completude1, _),
{Sem1=.. [Pred, Var|Args],
merge(Args, [Q, Gen, Num], Args1),
Sem=.. [Pred, Var|Args1],
interpretar(Sem, completo),
atualizar_pa(X, Gen, Num)}.
```

```
np(X, Ent, np(Adet, Arvore1, C_arvores), Gen, Num, [Ent|Entidades], Predicados,
```

```

Arg) -->
det(Q, Adet, Gen, Num),
nproprio(Slots1, Tipo, X, Sem1, Sem2, Arvore1, Gen, Num, Completude1, _, _),
completar(Completude1, Preds, Filler, C_arvores, Entidades, Predicados),
{Sem1=.. [P, Var|Args],
merge(Args, Filler, Args1),
merge(Args1, [Q, Gen, Num], Args2),
Ent=.. [P, Var|Args2],
interpretar(Ent, completo),
atualizar_pa(X, Gen, Num)}.

np(X, Sem1, Arvore, Gen, Num, [Sem1], [], Arg) -->
nproprio(Slots1, Tipo, X, Sem1, Sem2, Arvore, Gen, Num, Completude1, _, _),
{interpretar(Sem1, completo),
atualizar_pa(X, Gen, Num)}.

np(X, Entidade, NP_arvore, Gen, Num, [Entidade|Entidades2], Predicados2, Arg) -->
det(Q, Adet, Gen, Num),
adj(_, _, A_sem1, A_arvore1, Num, Nil1),
n(Slots, Tipo, X, Sem, N_arvore, Gen, Num, Completude, Ent1, Prop1),
adj(_, _, A_sem2, A_arvore2, Num, Nil2),
s(complemento, S_arvore, S_var, [np=X], Num, Entidades2, Predicados2),
{qualifiers(A_sem1, A_sem2, A_sem),
aplanar(A_sem, A_sem0),
(Sem=.. [Pred, X|Args]; Sem=.. [Pred], Args=[]),
merge(Args, A_sem0, Filler2),
merge(Filler2, [Q, Gen, Num], Filler3),
Entidade=.. [Pred, X|Filler3],
NP_arvore=.. [np, Adet, A_arvore1, N_arvore, A_arvore2, S_arvore],
instanciar_entidade(X),
Term=.. [Pred, X],
assert(Term),
assert(ativo(X)),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([Term, ativo(X)|L])),
atualizar_pa(X, Gen, Num)}.

np(X, Entidade, NP_arvore, Gen, Num, [Entidade|Entidades], Predicados, Arg) -->
det(Q, Adet, Gen, Num),
n(Slots, Tipo, X, Sem, N_arvore, Gen, Num, Completude, Ent1, Prop1),
completar(Completude, Preds, Filler, C_arvores, Entidades, Predicados),

```

```
{(Sem=.. [Pred,X|Args];Sem=.. [Pred],Args=[]),
merge(Args,Filler,Filler1),
merge(Filler1,[Q,Gen,Num],Filler2),
Entidade=.. [Pred,X|Filler2],
NP_arvore=.. [np,Adet,N_arvore|C_arvores],
interpretar(Entidade,completo),
atualizar_pa(X,Gen,Num)}.
```

```
np(X,Entidade,NP_arvore,Gen,Num,[Entidade],[],Arg) -->
det(Q,Adet,Gen,Num),
n(Slots,Tipo,X,Sem,N_arvore,Gen,Num,incompleto,Ent1,Prop1),
{(Sem=.. [Pred,X|Args];Sem=.. [Pred],Args=[]),
merge(Args,[Q,Gen,Num],Filler2),
Entidade=.. [Pred,X|Filler2],
NP_arvore=.. [np,Adet,N_arvore],
((Q==nodet;Q==8;Q==algun),!,
instanciar_entidade(X),
Term=.. [Pred,X],
assert(Term),
assert(ativo(X)),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([Term,ativo(X)|L])));
(Q==det;Q==dem1;Q==dem2;Q==dem3),
interpretar(Entidade,incompleto)),
atualizar_pa(X,Gen,Num)}.
```

```
np(X,Entidade,NP_arvore,Gen,Num,[Entidade],[],arg) -->
n(Slots,Tipo,X,Sem,N_arvore,Gen,Num,completo,Ent1,Prop1),
adj(_,,A_sem1,A_arvore1,Num,Nil1),
{(Sem=.. [Pred,X|Args];Sem=.. [Pred],Args=[]),
(A_sem1==[],!,merge(Args,[nodet,Gen,Num],Filler2);
merge(Args,[A_sem1,nodet,Gen,Num],Filler2)),
Entidade=.. [Pred,X|Filler2],
NP_arvore=.. [np,Adet,N_arvore,A_arvore1],
interpretar(Entidade,completo),
atualizar_pa(X,Gen,Num)}.
```

```
np(X,Entidade,NP_arvore,Gen,Num,[Entidade|Entidades],Predicados,arg) -->
adj(_,,A_sem1,A_arvore1,Num,Nil1),
n(Slots,Tipo,X,Sem,N_arvore,Gen,Num,Completude,Ent1,Prop1),
adj(_,,A_sem2,A_arvore2,Num,Nil2),
completar(Completude,Preds,Filler,C_arvores,Entidades,Predicados),
```

```

adj(,_,A_sem3,A_arvore3,Num,Nil3),
{qualifiers(A_sem1,A_sem2,A_sem0),
qualifiers(A_sem0,A_sem3,A_sem),
merge(A_sem,Filler,Filler1),
(Sem=..[P,X|Resto];Sem=..[P],Resto=[]),
merge(Resto,Filler1,Filler2),
merge(Filler2,[none,Gen,Num],Filler3),
Entidade=..[P,X|Filler3],
NP_arvore=..[np,A_arvore1,N_arvore,A_arvore2,A_arvore3|C_arvores],
interpretar(Entidade,completo),
atualizar_pa(X,Gen,Num)}.

```

```

np(X,modificador=Sem,N_arvore,Gen,Num,[],[],noarg) -->
n(Slots,Tipo,X,Sem,N_arvore,Gen,Num,Completude,Ent,Pro).

```

```

np(X,modificador=Entidade,NP_arvore,Gen,Num,Entidades,Predicados,noarg) -->
adj(,_,A_sem1,A_arvore1,Num,Nil1),
n(Slots,Tipo,X,Sem,N_arvore,Gen,Num,Completude,Ent1,Prop1),
completar(Completude,Preds,Filler,C_arvores,Entidades,Predicados),
adj(,_,A_sem2,A_arvore2,Num,Nil2),
{qualifiers(A_sem1,A_sem2,A_sem),
merge(A_sem,Filler,Filler1),
merge(Filler1,[none,Gen,Num],Filler2),
Entidade=..[Sem|Filler2],
NP_arvore=..[np,A_arvore1,N_arvore,A_arvore2|C_arvores]}}.

```

```

completar(completo,[],[],[],[],[]) --> [].

```

```

completar(incompleto,Preds,Slots,[Arvore|Arvores],Entidades,Predicados) -->
possibel_complemento,
pp(X,Pred,Arvore,Entidades1,Predicados1,noarg),
completar(incompleto,Preds2,Slots2,Arvores,Entidades2,Predicados2),
{merge(Entidades1,Entidades2,Entidades),
merge(Predicados1,Predicados2,Predicados),
(modificador(Pred),!,Preds=Preds2,Slots=[Pred|Slots2];
Preds=[Pred|Preds2],Slots=[especificador=X|Slots2])}.

```

```

completar(incompleto,[],[],[],[],[]) --> [],!.

```

```

possibel_complemento([prep(de,Tipo,Termo)|L],[prep(de,Tipo,Termo)|L]).

```

```

possibel_complemento([prep(em,Tipo,Termo)|L],[prep(em,Tipo,Termo)|L]).

```

```

possibel_complemento([prep(sobre,Tipo,Termo)|L],[prep(sobre,Tipo,Termo)|L]).

possibel_complemento([prep(por,Tipo,Termo)|L],[prep(por,Tipo,Termo)|L]).

modificador(modificador=T).

qualifiers([],[],[]):-!.

qualifiers([],T,[T]):-!.

qualifiers(T,[],[T]):-!.

qualifiers(T1,T2,[T1,T2]):-!.

completar1([],[],[],[],[],Tipo_verbo,[],V_var) --> [].

adj(Slots,Var,Sem,a(A),Num,nonil) -->
[a(A,T,Sem,Num)],!.

adj(Slots,Var,[],[],Num,nil) --> [].

fill([],[],[],Num,[],[]) -->
[],!.

fill(Slots,SF,AS,Num,Entidades,Predicados,L1,L3):-
select(Slot,Slots,Slots1),
fillone(Slot,S,Aslot,Num,Entidades1,Predicados1,L1,L2),
fill(Slots1,R,Aslots,Num,Entidades2,Predicados2,L2,L3),
merge(Entidades1,Entidades2,Entidades),
merge(Predicados1,Predicados2,Predicados),
(S==[],SF=R;SF=[S|R]),(Aslot==[],AS=Aslots;
AS=[Aslot|Aslots]).

fillone([Tipo,s(T)],Tipo=Var,Arvore,Num,Entidades,Predicados) -->
s(T,Arvore,Var,_,Entidades,Predicados).

fillone([Tipo,pp(X)],Tipo=Y,pp(Parvore,Arvore),_,Entidades,Predicados,
[prep(X,TipoPrep,Parvore)|L],L1):-
np(Y,Entidade,Arvore,Gen,Num,Entidades,Predicados,arg,L,L1).

```

```
fillone([Tipo,np],Tipo=X,Arvore,_,Entidades,Predicados,L,L1):-
np(X,Entidade,Arvore,Gen,Num,Entidades,Predicados,arg,L,L1).
```

```
n(Slots,Tipo,Var,Sem,n(Var,N),Gen,Num,Completude,[Var],[]) -->
[nome(N,Tipo,Slots,Completude,Sem,Gen,Num)].
```

```
nproprio(Slots,Tipo,Var,Sem1,Sem2,nome_proprio(Var,N),Gen,Num,
Completude,[Var],[]) -->
[nome_prop(N,Tipo,Slots,Completude,Var,Sem1,Sem2,Gen,Num)].
```

```
det(X,det(D),Gen,Num) -->
[determinante(D,X,Gen,Num)].
```

```
select(A,[A|L],L).
select(A,[B|L],[B|L1]):-select(A,L,L1).
```

```
aplanar([],[]).
aplanar([A],B):-lista(A),aplanar(A,B),!.
aplanar([A],[A]).
aplanar([[A1|A2]|Resto],Resultado):-
aplanar([A1|A2],B),aplanar(Resto,Resto1),
merge(B,Resto1,Resultado),!.
aplanar([A|Resto],[A|Resultado]):-aplanar(Resto,Resultado).
```

```
lista([]).
lista([_|_]).
```

```
member(X,[X|_]).
member(X,[_|L]):-member(X,L).
```

```
merge([],A,A).
merge([A|Resto],B,[A|RestoB]):-merge(Resto,B,RestoB).
```

```
multi_merge([],[]).
```

```

multi_merge([Lista|Listas],Resultado):-
multi_merge(Listas,Resultado1),
merge(Lista,Resultado1,Resultado).

segundo([A,B|_],B).

ultimo([A],A).

ultimo([A|R],B):-ultimo(R,B).

instanciar_entidade(Var):-(atom(Var);retract(ultimo_entidade(X)),
Y is X+1,
name(X,L),
name(Var,[120|L]),
assert(ultimo_entidade(Y))).

instanciar_evento(Var):-retract(ultimo_evento(X)),
Y is X+1,
name(X,L),
name(Var,[112|L]),
assert(ultimo_evento(Y)).

instanciar_pronome(Pro):-retract(ultimo_pronome(X)),
Y is X+1,
name(X,L),
name(Pro,[105|L]),
assert(ultimo_pronome(Y)).

iniciar_entidade:-(ultimo_entidade(X),retract(ultimo_entidade(X));true),
assert(ultimo_entidade(1)).

iniciar_evento:-(ultimo_evento(X),retract(ultimo_evento(X));true),
assert(ultimo_evento(1)).

iniciar_pronome:-(ultimo_pronome(X),retract(ultimo_pronome(X));true),
assert(ultimo_pronome(1)).

read_word1(39,[],X):-get0(X).

read_word1(Char,[Char|Chars],Last):-

```

```
get0(Next),
read_word1(Next,Chars,Last).
```

```
read_word(C, [],C):- (espaco(C);virgula(C);ponto_virgula(C);ponto(C)),!.
read_word(C, [],C):-nl(C),!.
```

```
read_word(Char, [Char|Chars],Last):-
get0(Next),
read_word(Next,Chars,Last).
```

```
read_sent(Words):-
get0(Char),
(Char=95,Words=[];
read_sent(Char,Words)).
```

```
read_sent(C, []):-nl(C),!.
```

```
read_sent(C,Words):- (espaco(C);ponto_virgula(C);virgula(C);ponto(C)),!,
get0(Char),
read_sent(Char,Words).
```

```
read_sent(Char, [Word|Words]):-
(Char=39,get0(Char1),read_word1(Char1,Chars,Next);
read_word(Char,Chars,Next)),
name(Word,Chars),
read_sent(Next,Words).
```

```
nova_línea(10).
espaco(32).
ponto(46).
nl(10).
```

```
% particionamento do texto
```

```
objetivo(I):-
segmento(I,0),
predicadores(objetivo,P),
conhecidas(C),
member(autor(X),C),
member(artigo(Y),C),!,
verificar(0,P,X,Y).
```

```

objetivo(I):-
segmento(I,O),
predicadores(objetivo,P),
verificar(O,P).

```

```

predicadores(objetivo,[objetivo,apresentar,relatar,mostrar]).

```

```

metodologia(I):-
segmento(I,O),
predicadores(metodologia,P),
verificar(O,P).

```

```

predicadores(metodologia,[selecao,corpo_teorico]).

```

```

contexto(I):-
segmento(I,O),
predicadores(contexto,P),
verificar(O,P).

```

```

predicadores(contexto,[necessaria,problema,dificuldade]).

```

```

conhecidas([trabalho(e6),artigo(e6),autor(e5)]).

```

```

predicadores(objetivo,[apresentar,relatar,mostrar]).

```

```

conclusao(I):-
sugestao(I).

```

```

conclusao(I):-
recomendacao(I).

```

```

sugestao(I):-
segmento(I,O),
predicadores(sugestao,P),
verificar(O,P).

```

```

verificar(O,[Predicado|R]):-
verificar(O,Predicado),!.

```

```

verificar(O,[Predicado|R]):-
verificar(O,R).

```

```

verificar([P|R],Predicado):-
(P=..[Predicado|Argumentos];
verificar(R,Predicado)).

```

```

predicadores(sugestao,[sugerir,sugestao]).

```

```

recomendacao(I):-
segmento(I,O),
predicadores(recomendacao,P),
verificar(O,P).

```

```

predicadores(recomendacao,[recomendar,recomendacao]).

```

```

contexto(I):-
segmento(I,S),
predicadores(background,P),
verificar(S,P).

```

```

predicadores(background,[necessario]).

```

```

experimento(I):-
segmento(I,S),
predicadores(experimento,P),
verificar(S,P).

```

```

predicadores(experimento,[experimento,experiencia]).

```

```

busca_ref(Gen,Num,Ref):-
pilha(PA),
busca_ref(Gen,Num,Ref,PA).

```

```

busca_ref(Gen,Num,Ref,[[Gen,Num,Ref1]|Resto]):-
Ref=Ref1.

```

```

busca_ref(Gen,Num,Ref,[[Gen1,Num1,Ref1]|Resto]):-
busca_ref(Gen,Num,Ref,Resto).

```

```

atualizar_pa(Ref,Gen,Num):-
pilha(PA),
retract(pilha(PA)),

```

```
(PA=[Tope|Resto],Tope==[Gen,Num,Ref],PN=PA;
PN=[[Gen,Num,Ref]|PA]),
assert(pilha(PN)).
```

```
iniciar_pa:-
(retract(pilha(PA));true),
assert(pilha([])).
```

```
interpretar(Expressao,completo):-
referente(Expressao).
```

```
interpretar(Expressao,incompleto):-
co-referente(Expressao).
```

```
disciplina(e1,nome=calculo_numerico):-
assert(disciplina(e1)),
assert(ativo(e1)),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([disciplina(e1),ativo(e1)|L])).
```

```
plano(p(e1),modificador=ensino,especificador=e1).
```

```
metodologia(m(e1),modificador=ensino,especificador=e1).
```

```
disciplina(e4,nome=similitude_em_engenharia):-
assert(disciplina(e4)),
assert(ativo(e4)),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([disciplina(e4),ativo(e4)|L])).
```

```
metodologia(m(X),modificador=ensino):-
disciplina(X),
ativo(X),
assert(metodologia(m(X))),
assert(ativo(m(X))),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([metodologia(m(X)),ativo(m(X))|L])).
```

```
plano(p(X),modificador=ensino):-
disciplina(X),
ativo(X),
assert(plano(p(X))),
assert(ativo(p(X))),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([plano(p(X)),ativo(p(X))|L])).
```

```
ciclo(c(X),modificador=aulas):-
disciplina(X),
ativo(X),
assert(ciclo(c(X))),
assert(ativo(c(X))),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([ciclo(c(X)),ativo(c(X))|L])).
```

```
universidade(e7,nome=ufu):-
assert(universidade(e7)),
assert(ativo(e7)),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([universidade(e7),ativo(e7)|L])).
```

```
curso(c(e7),modificador=engenharia_mecanica,especificador=e7).
```

```
teoria(e8,modificador=modelo):-
assert(teoria(e8)),
assert(ativo(e8)),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([teoria(e8),ativo(e8)|L])).
```

```
fundamento(f(X)):-
teoria(X),
ativo(X).
```

```
fundamento(f(e8),especificador=e8).
```

```
area(e9,nome=mecanica_dos_fluidos):-
assert(area(e9)),
assert(ativo(e9)),
```

```
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([area(e9),ativo(e9)|L])).

aluno(a(X)):-
disciplina(X),
ativo(X).

curso(c(X),modificador=engenharia_mecanica):-
universidade(X),
ativo(X).

autor(e5).

artigo(e6).

trabalho(e6).

engenharia(e2).

ciencia(e3).

trocador(e9,modificador=calor):-
assert(trocador(e9)),
assert(ativo(e9)),
lista_ativos(L),
retract(lista_ativos(L)),
assert(lista_ativos([trocador(e9),ativo(e9)|L])).

tipo(e9,modificador=trocador):-
trocador(e9),
ativo(e9).

ativo(e6).

aluno(e10).

dado(e11).

prototipo(e12).

modelo(e13).
```

```
quantidade(e14).

calculador(e15).

engenheiro(e16).

queda_pressao(e17).

elemento(e18,qualificador=basico).

estudante(e19).

criatividade(e20).

relacionado(X,Y):-
Y=.. [F,X].

relacionado(X,Y):-
X=.. [F,Y].

referente(T):-
referente(T,Ref).

referente(T,Var):-
T=.. [P,Var|Args],
tirar_determinado(Args,Args1),
T1=.. [P,Var|Args1],
call(T1).

tirar_determinado([D,G,N],_):-!.

tirar_determinado([X|R],[X|Res]):-tirar_determinado(R,Res).

nuevos(T):-
T=.. [P,V|R],
T1=.. [P,V],
(call(T1);
assert(T1)).
```

```

modificadores([especificador=X|Resto],Resto1):-
modificadores(Resto,Resto1).

modificadores([modificador=X|Resto],[modificador=X|Resto1]):-
modificadores(Resto,Resto1).

modificadores([nome=X|resto],[nome=X|resto1]):
modificadores(Resto,Resto1).

modificadores([],[]).

especificador([especificador=X|Resto],X).

especificador([Y=X|Resto],E):-
especificador(Resto,E).

co-referente(T):-
T=..[P,V|R],
T1=..[P,V],
call(T1),call(ativo(V)).

referentes:-
referentes(1).

referentes(I):-
descricoes(I,D),
referentes(D,R),
assert(ref(I,R)),
atualizar_tabla(I,R),
J is I+1,
referentes(J).

referentes(_).

referentes([],[]).

referentes([Pred|Resto],[Ref|RestoRef]):-
Pred=..[P,Ref|_],
referentes(Resto,RestoRef).

atualizar_tabla(I,[]).

```

```

atualizar_tabla(I, [Ref|Resto]):-
antecedente(Ref,Ref1),
(tabla(Ref1,J);assert(tabla(Ref1,I))),
atualizar_tabla(I,Resto).

```

```

antecedente(Ref,Ref):-atom(Ref).

```

```

antecedente(Ref,Ref1):-Ref=.. [F,Ref1].

```

```

relacion(I,elabora(J)):-
ref(I,Referentes),
member(Ref,Referentes),
antecedente(Ref,Ref1),
tabla(Ref1,J),\+I=J.

```

```

relacion(I,nada).

```

```

apagar_referentes:-
retract(ref(I,R)),fail.

```

```

apagar_referentes:-
retract(tabla(I,E)),fail.

```

```

apagar_referentes.

```

```

apagar_arvore2:-
retract(arvore_coesivo(I,S)),fail.

```

```

apagar_arvore2.

```

```

apagar_pilhas:-
retract(pila_sentencas(X)),
retract(pila_relacoes(Y)),
retract(pila_coesiva(Z)),
retract(pilha(W)).

```

```

apagar_categoria:-
retract(categoria_atual(X)).

```

```

apagar_sumario:-
retract(sumario(S)).

```

```

apagar_auxiliares2:-
apagar_referentes,
apagar_pilhas,
apagar_categoria,
apagar_sumario.

```

```

apagar_auxiliares2.

```

```

iniciar_la:-
assert(lista_ativos([])).

```

```

apagar_ativados:-
lista_ativos(L),
apagar_ativos(L).

```

```

apagar_ativados.

```

```

apagar_ativos([]):-
retract(lista_ativos(L)).

```

```

apagar_ativos([E|R]):-
retract(E),
apagar_ativos(R).

```

```

analisar_coesao:-
assert(pila_sentencas([1])),
assert(pila_relacoes([])),
analisar_coesao(2).

```

```

analisar_coesao(I):-
relacion(I,E),
pila_sentencas(P),
pila_relacoes(R),
retract(pila_sentencas(P)),
retract(pila_relacoes(R)),
assert(pila_sentencas([I|P])),
assert(pila_relacoes([E|R])),
J is I+1,
analisar_coesao(J).

```

```

analisar_coesao(_).

```

```

iniciar_analisis:-
assert(pila_sentencas([1])),
assert(pila_relacoes([])),
assert(pila_coesiva([])),
classificar(1,Classe),
(Classe==nada,assert(categoria_atual(contexto));
assert(categoria_atual(Classe))),
assert(sumario([])).

```

```

finalizar_analisis1:-
fechar,
pila_coesiva(PSC),
(PSC=[Uno],Res=Uno;Res=..[sequencia|PSC]),
sumario(Sum),
retract(sumario(Sum)),
categoria_atual(Ca),
Segmento=..[Ca,Res],
agregar(Segmento,Sum,NewSum),
assert(sumario(NewSum)).

```

```

agregar(Seg, [], [Seg]).

```

```

agregar(Seg, [P|R], [P|R1]) :- agregar(Seg, R, R1).

```

```

reversa([], []).

```

```

reversa([A|B], B1) :-
reversa(B, B2),
agregar(A, B2, B1).

```

```

fechar:-
pila_sentencas([S]),
pila_relacoes([]),
pila_coesiva(PSC),
retract(pila_coesiva(PSC)),
assert(pila_coesiva([S|PSC])).

```

```

um_passo(I):-
classificar(I,C),
categoria_atual(Ca),
((C==Ca;C==nada),atualizar1(I);
reduzir_ate_um,
pila_sentencas([Uno]),
retract(pila_sentencas([Uno])),

```

```

assert(pila_sentencas([I]),
pila_relacoes([]),
pila_coesiva(PSC),
retract(pila_coesiva(PSC)),
assert(pila_coesiva([])),
(PSC==[],Res=..[Ca,Uno];
Res=..[Ca,Uno|PSC]),
sumario(Sum),
retract(sumario(Sum)),
agregar(Res,Sum,NewSum),
assert(sumario(NewSum)),
retract(categoria_atual(Ca)),
assert(categoria_atual(C))).

```

```

atualizari(I):-
relacion(I,E),
pila_sentencas(S),
pila_relacoes(R),
(S=[Uno],
retract(pila_sentencas(S)),
retract(pila_relacoes(R)),
% se 'e uma elaboracao da sentenca na pilha
(E==elabora(Uno),
assert(pila_sentencas([I|S])),
assert(pila_relacoes([E]));
% se nao elabora a sentenca na pilha
% retirar o segmento da pilha e coloca-lo na pilha de segmentos
pila_coesiva(PSC),
retract(pila_coesiva(PSC)),
assert(pila_coesiva([Uno|PSC])),
assert(pila_sentencas([I])),
assert(pila_relacoes([])))
;
% tem mais do que um elemento
% e nao tem vinculo
E==nada, reduzir_ate_um,
pila_sentencas([NewS]),
pila_relacoes([]),
retract(pila_sentencas([NewS])),
retract(pila_relacoes([])),
assert(pila_sentencas([I])),
assert(pila_relacoes([])),
pila_coesiva(PSC),

```

```

retract(pila_coesiva(PSC)),
assert(pila_coesiva([NewS|PSC]));
% tem mais do que um elemento
% e um paralelo
tope(S,TS),
tope(R,TR),
E==TR,
S1=[I|S],R1=[E|R],
retract(pila_sentencas(S)),
retract(pila_relacoes(R)),
assert(pila_sentencas(S1)),assert(pila_relacoes(R1));
% tem mais do que um elemento
% e uma elaboracao da sentenca no topo da pilha
tope(S,TS),
tope(R,TR),
elabora(TS)=E,
S1=[I|S],
R1=[E|R],
retract(pila_sentencas(S)),
retract(pila_relacoes(R)),
assert(pila_sentencas(S1)),assert(pila_relacoes(R1));
% a nova sentenca elabora um elemento que esta no interior da pilha
reducao(E),
pila_sentencas(S),
pila_relacoes(R),
retract(pila_sentencas(S)),
retract(pila_relacoes(R)),
assert(pila_sentencas([I|S])),
assert(pila_relacoes([E|R]));
reduzir_ate_um).

reducao(elabora(X)):-
pila_sentencas([Uno]).

reducao(elabora(X)):-
pila_relacoes([elabora(X)|Resto]).

reducao(elabora(X)):-
reduzir,
reducao(elabora(X)).

reduzir_ate_um:-
pila_sentencas([Uno]).

```

```

reduzir_ate_um:-
pila_sentencas([Tope|Resto]),
reduzir.

```

```

tope([A|B],A).

```

```

reduzir:-
pila_sentencas(S),
pila_relacoes(R),
reduzir(R,N),
N1 is N + 1,
tirar_rel(R,N,R1),
tirar_sen(S,N1,Principio,Resto),
last(Principio,Last,Primeiras),
retract(pila_sentencas(S)),
retract(pila_relacoes(R)),
(N > 1,
reversa(Primeiras,RevPrimeiras),
Paralelo=..[paralelo|RevPrimeiras],
Elaboracao=..[elaboracao,Last,Paralelo];
N==1,
reversa(Principio,Principio1),
Elaboracao=..[elaboracao|Principio1]),
assert(pila_relacoes(R1)),
assert(pila_sentencas([Elaboracao|Resto])).

```

```

last([A],A,[]).

```

```

last([A|R],L,[A|R1]):-
last(R,L,R1).

```

```

tirar_rel(R,0,R).

```

```

tirar_rel([Rel|Rels],N,Rels1):-
N >= 1,
N1 is N - 1,
tirar_rel(Rels,N1,Rels1).

```

```

tirar_sen(S,0,[],S).

```

```

tirar_sen([Sen|Sents],N,[Sen|Prin],Resto):-
N >= 1, N1 is N - 1,
tirar_sen(Sents,N1,Prin,Resto).

```

```
reduzir([],0).
```

```
reduzir([A|R],N):-
  contar(A,[A|R],N).
```

```
contar(A,[],0).
```

```
contar(A,[X|R],0):-(!+A==X).
```

```
contar(A,[A|R],N):-contar(A,R,M),N is M + 1.
```

```
ver_estado:-
  pila_sentencas(S),
  pila_relacoes(R),
  pila_coesiva(PSC),
  categoria_atual(Ca),
  sumario(Sum),
  write('Sentencas => '),write(S),nl,
  write('Relacoes => '),write(R),nl,
  write('Segmentos => '),write(PSC),nl,
  write('Categoria => '),write(Ca),nl,
  write('Sumario => '),write(Sum).
```

```
limpar_estado:-
  retract(pila_sentencas(S)),
  retract(pila_relacoes(R)),
  retract(pila_coesiva(C)),
  retract(sumario(Sum)),
  retract(categoria_atual(Cat)).
```

```
classificar(S,contexto):-
  nao_fechada(contexto),contexto(S).
```

```
classificar(S,objetivo):-
  nao_fechada(objetivo),objetivo(S).
```

```
classificar(S,conclusao):-
  nao_fechada(conclusao),conclusao(S).
```

```
classificar(S,recomendacao):-
  nao_fechada(recomendacao),recomendacao(S).
```

```

classificar(S,sugestao):-
nao_fechada(sugestao),sugestao(S).

```

```

classificar(S,metodologia):-
nao_fechada(metodologia),metodologia(S).

```

```

classificar(S,experimento):-
nao_fechada(experimento),experimento(S).

```

```

classificar(S,resultado):-
nao_fechada(resultado),resultado(S).

```

```

classificar(S,nada).

```

```

nao_fechada(contexto).
nao_fechada(objetivo).
nao_fechada(recomendacao).
nao_fechada(sugestao).
nao_fechada(metodologia).
nao_fechada(experimento).
nao_fechada(resultado).
nao_fechada(conclusao).

```

```

co_sentenca(I,S1):-
segmento(I,S),
pro_coerencia(S,S1).

```

```

pro_coerencia([P|R],[P1|R1]):-
mudar_predicado(P,P1),!,
pro_coerencia(R,R1).

```

```

pro_coerencia([P,Q|R],[P,E|R1]):-
elabora(P,Q,E),
pro_coerencia([Q|R],R1).

```

```

pro_coerencia([P|R],[P1|R1]):-
classificar_sequencia(P,R,P1),
pro_coerencia(R,R1).

```

```

pro_coerencia([P|R],[P|R1]):-
pro_coerencia(R,R1).

```

```

tem_elabora([elaboracao|R]):-!.

```

```

tem_elabora([P|R]):-tem_elabora(R).

```

```

pro_coerencia(R,R).

```

```

mudar_predicado(P,P1):-
P=..[objetivo,Arg|Resto],
P1=..[objetivo|Resto].

```

```

elaborap(Pro1,Pro2,E):-
Pro1=..[Pred1,Var1|Resto1],
Pro2=..[Pred2,Var2|Resto2],
(\+Pred1=oposicao,\+Pred1=causa_consequencia,\+Pred1=necessario),
member(Var2,Resto1),
E=..[elaboracao].

```

```

elabora(Pro1,Pro2,E):-
Pro1=..[Pred1,Var1|Entidades1],
Pro2=..[Pred2,Var2|Entidades2],
intersecao(Entidades1,Entidades2),
E=..[elaboracao,Var1,Var1,Var2].

```

```

construir_arvore(I,A):-
segmento(I,S),
tirar_entidades(S,S1),
pro_coerencia(S1,S2),
construir(S2,A),
assert(arvore_coesivo(I,A)).

```

```

construir_sumario:-
sumario(S),
reemplazar(S,S1),
retract(sumario(S)),
assert(sumario(S1)).

```

```

reemplazar([Cat|Cats],[Cat1|Cats1]):-
Cat=..[Nome|Termo],
reem(Termo,Termo1),

```

```
Cat1=..[Nome|Termo1],
reemplazar(Cats,Cats1).
```

```
reemplazar([],[]).
```

```
reem(Term,Term2):-
number(Term),!,
arvore_coectivo(Term,[Term1]),
retirar_evento(Term1,Term2).
```

```
retirar_evento(Term,Term1):-
Term=..[Pred|Terms],
retirar_eventos(Terms,Term1),
Term1=..[Pred|Term1].
```

```
retirar_eventos([Evento|Resto],Resto1):-
evento(Evento),!,
retirar_eventos(Resto,Resto1).
```

```
retirar_eventos([NoEvento|Resto],[NoEvento1|Resto1]):-
(NoEvento=..[Pred,Var|Terms],retirar_evento(NoEvento,NoEvento1);
NoEvento1=NoEvento),
retirar_eventos(Resto,Resto1).
```

```
retirar_eventos([],[]).
```

```
reem(Term,Term1):-
Term=..[Pred|Termos],
reems(Termos,Term1),
Term1=..[Pred|Term1].
```

```
reems([],[]).
```

```
reems([Term|Terms],[Term1|Term1]):-
reem(Term,Term1),
reems(Terms,Term1).
```

```
tirar_entidades([Prop|Props],[Prop|Prop1]):-
```

```

Prop=.. [Pred,Var|Resto],
evento(Var),!,
tirar_entidades(Props,Props1).

tirar_entidades([Prop|Props],Props1):-
tirar_entidades(Props,Props1).

tirar_entidades([],[]).

evento(Var):-
name(Var,[112|R]).

construir([Prop],[Prop]).

construir([Prop|Props],Props1):-
terminais([Prop|Props],Terms,NoTerms),
aplicar(Terms,NoTerms,Arvores),
construir(Arvores,Props1).

aplicar([],NoTerms,NoTerms).

aplicar([Term|Terms],NoTerms,Arvores):-
aplicar_termo(Term,NoTerms,NoTerms1),
aplicar(Terms,NoTerms1,Arvores).

% aplica um terminal a o seu pai

aplicar_termo(Term,[NoTerm|NoTerms],[NoTerm1|NoTerms1):-
Term=.. [Pred1,Var1|Vars1],
NoTerm=.. [Pred2,Var2|Vars2],
member(Var1,Vars2),
incluir(NoTerm,Term,NoTerm1).

aplicar_termo(Term,[NoTerm|NoTerms],[NoTerm|NoTerms1):-
aplicar_termo(Term,NoTerms,NoTerms1).

incluir(Prop1,Prop2,Res):-
Prop1=.. [Pred1,Var1|Vars1],
Prop2=.. [Pred2,Var2|Vars2],
partir(Vars1,Var2,Principio,Fim),
merge(Principio,[Prop2],Principio1),
merge(Principio1,Fim,NewVars1),
Res=.. [Pred1,Var1|NewVars1].

```

```
partir([A|B],A, [],B):-!.

```

```
partir([A|B],C,[A|R1],R2):-
partir(B,C,R1,R2).

```

```
terminais([Prop|Props],[Prop|Props1],Props2):-
terminal(Prop),!,
terminais(Props,Props1,Props2).

```

```
terminais([Prop|Props],Props1,[Prop|Props2]):-
terminais(Props,Props1,Props2).

```

```
terminais([],[],[]).

```

```
terminal(Prop):-
Prop=..[Pred,Var|Vars],
var_terminais(Vars).

```

```
var_terminais([]).

```

```
var_terminais([Var|Vars]):-
var_terminal(Var),
var_terminais(Vars).

```

```
var_terminal(Var):-
name(Var,[120|R]).

```

```
var_terminal(Var):-
name(Var,[101|R]).

```

```
var_terminal(Var):-
Var=..[F,Vars|R].

```

```
trazer_associados([],[],Predicados).

```

```
trazer_associados([Var|Vars],[Pred|Preds],Predicados):-
buscar(Var,Predicados,Pred),
trazer_associados(Vars,Preds,Predicados).

```

```
buscar(Var,[Pred|Preds],Pred):-
Pred=..[P,Var|Vars].

```

```

buscar(Var, [Pred|Preds], Pred1):-
buscar(Var, Preds, Pred1).

intersecao([A|B], C):-
(member(A, C);
intersecao(B, C)).

clasificar_sequencia(S, S1, Props):-
S=.. [sequencia, Var|Vars],
conjuntos(Vars, Props, Conjuntos),
intersecar(Conjuntos, C),
(C=[], S1=S; S1=.. [paralelo, Var|Vars]).

conjuntos([], Props, []).

conjuntos([Var|Vars], Props, [CVar|CVars]):-
buscar_prop(Var, Props, Prop),
Prop=.. [Pred, Arg|CVar],
conjuntos(Vars, Props, CVars).

buscar_prop(Var, [Prop|Props], Prop):-
Prop=.. [Pred, Var|Resto].

buscar_prop(Var, [Prop|Props], Prop1):-
buscar_prop(Var, Props, Prop1).

intersecar([C], C).

intersecar([Conj|Conjs], R):-
intersecar(Conjs, R1),
intersec(Conj, R1, R).

intersec([], B, []).

intersec([A|B], C, R):-intersec1([A|B], C, R).

intersec1([], C, []).

intersec1([A|B], C, [A|B1]):-
member(A, C), !,
intersec1(B, C, B1).

```

```
intersec1([A|B],C,B1):-
intersec1(B,C,B1).
```

```
imprimir_sumario:-
sumario(S),
write('COMEÇO SUMARIO'),nl,
imprimir_sumario(S).
```

```
imprimir_sumario([]):-
write('FIM SUMARIO'),nl.
```

```
imprimir_sumario([Cat|Cats]):-
imprimir_categoria(Cat),nl,
imprimir_sumario(Cats).
```

```
imprimir_categoria(Cat):-
Cat=..[Categoria|Termos],
imprimir_nome_cat(Categoria),
imprimir_relacoes(Termos).
```

```
imprimir_relacoes([]).
```

```
imprimir_relacoes([Rel|Rels]):-
imprimir_relacao(Rel),
imprimir_relacoes(Rels).
```

```
imprimir_relacao(Rel):-
Rel=..[Relacao|Termos],
(relacao(Relacao),nl,imprimir_nom_rel(Relacao),imprimir_termos(Termos);
write(Rel),nl).
```

```
imprimir_termos([Term|Terms]):-
Term=..[Pred|Vars],
(relacao(Pred),nl,imprimir_nom_rel(Pred),imprimir_relacoes(Vars);
write(Term),nl),
imprimir_termos(Terms).
```

```
imprimir_termos([]):-nl.
```

```
relacao(elaboracao).
relacao(paralelo).
relacao(sequencia).
```

relacao(ocasio).

relacao(causa_consequencia).

imprimir_nom_rel(elaboracao):-write('Elaboracao'),nl.

imprimir_nom_rel(paralelo):-write('Paralelo'),nl.

imprimir_nom_rel(sequencia):-write('Sequencia'),nl.

imprimir_nom_rel(ocasio):-write('Ocasiao'),nl.

imprimir_nom_rel(causa_consequencia):-write('Causa/Consequencia'),nl.

imprimir_nome_cat(objetivo):-write('OBJETIVO'),nl.

imprimir_nome_cat(metodologia):-write('METODOLOGIA'),nl.

imprimir_nome_cat(conclusao):-write('CONCLUSAO'),nl.

imprimir_nome_cat(sugestao):-write('SUGESTAO'),nl.

imprimir_nome_cat(recomendacao):-write('RECOMENDACAO'),nl.

imprimir_nome_cat(experimento):-write('EXPERIMENTO'),nl.

imprimir_nome_cat(contexto):-write('CONTEXTO'),nl.

Bibliografia

- [ABNT 1987] ABNT-Associação Brasileira de Normas Técnicas. *Resumos*. 1987.
- [Allen 1987] Allen, J. *Natural Language Understanding*. The Benjamin Cummings Publishing Company Inc., 1987.
- [Bazzo and Pereira, 1989] Bazzo, W.A. and Teixeira do Vale Pereira, L. *Criatividade na Engenharia*. Rev. Ensino Eng., São Paulo, 8(1):8-11, 1^o semestre 1989.
- [Borzani 1989] Borzani, W. *Definições e Cálculos de Velocidades Específicas em Processos Fermentativos*. Rev. Ensino Eng., São Paulo, 8(1):39-47, 1^o sem. 1989.
- [Brown e Yule 1983] Brown, G. e Yule, G. *Discourse Analysis*. Cambridge University Press, 1983.
- [Cleveland e Cleveland 1983] Cleveland, D.B. y Cleveland A.D. *Introduction to Indexing and Abstracting*. Libraries Unlimited, Inc., 1983.
- [Crystal 1987] Crystal, D. *The Cambridge Encyclopedia of Language*. Cambridge University Press, 1987.
- [Danlos 1987] Danlos, L. *The Linguistic Basis of Text Generation*. Studies in Natural Language Processing, Cambridge University Press, 1987.
- [da Silveira Neto e Hernandez Mendoza 1988] da Silveira Neto, A. e Hernandez Mendoza, O.S. *Trocadores de Calor Compactos -Bancada de Testes*. Rev. Ensino Eng., São Paulo, 7(1):43-48, 1^o Sem. 1988.
- [de Araújo e Szeremeta 1985] de Araújo, N.D. e Szeremeta, J.F. *Uma Experiência no Ensino de Cálculo Numérico na UFSC*. Rev. Ensino Eng., 4(2):138-139, São Paulo, 2^o Sem. 1985.
- [de Mello 1989] de Mello e Souza, J. *Universidade, Ciência e Aplicação do Conhecimento no Mundo Moderno*. Rev. Ensino Eng., São Paulo, 8(1):23-27, 1^o sem. 1989.
- [de Oliveira e Aita 1986] de Oliveira, B.T. e Aita, T.J. *O Ensino de Desenho no 1^o e 2^o graus e seus Reflexos no Ensino Superior*. Rev. Ensino Eng., São Paulo, 5(2):210-222, 2^o sem. de 1986.

- [de Sá Weyne 1987] de Sá Weyne, G.R. *O Ensino de Projetos de Indústrias Químicas no "University College" de Londres, Inglaterra. Um Estudo Comparativo com as Condições Brasileiras*. Rev. Ensino Eng., São Paulo, 6(1):85-96, 1^o sem. de 1987.
- [de Souza e Silva e Koch 1987] de Souza e Silva, M.C.P. e Koch, I.V. *Linguística Aplicada ao Português: Sintaxe*. Cortez Editora, 1987.
- [Eik et al. 1985] Eick, N.C., Schuck, M.T.G. de O. e Lisboa, N.A. *Sistema de Projeção Estereoscópica com Luz Polarizada*. Rev. Ensino Eng., São Paulo, 4(2):133-134, 2^o sem. 1985.
- [Forum ABENGE 1985] Forum ABENGE. *A Inovação Tecnológica e o Papel da Universidade*. Rev. Ensino Eng., São Paulo, 4(2):133-134, 2^o sem. 1985.
- [Gopnik 1972] Gopnik, M. *Linguistic Structures in Scientific Texts*. Mouton, 1972.
- [Grimes 1975] Grimes, J. *The Thread of Discourse*. Mouton and Company, The Hague, Netherlands, 1975.
- [Grosz e Sidner 1986] Grosz, B.J. e Sidner, C.L. *Attention, Intentions and the Structure of Discourse*. Computational Linguistics, Vol. 12, Num. 3, July-September 1986.
- [Halliday e Hasan 1976] Halliday, M.A. e Hasan, R. *Cohesion in English*. London, Longman Press, 1976.
- [Hatim e Mason 1990] Hatim, B. e Mason, I. *Discourse and the Translator*. Longman, 1990.
- [Hirst 1981] Hirst, G. *Anaphora in Natural Language Understanding: A Survey*. Lecture Notes in Computer Science 119. Springer-Verlag, 1981.
- [Hirschberg e Litman 1993] Hirschberg, J. e Litman, D. *Empirical Studies on the Disambiguation of Cue Phrases*. Computational Linguistics, Vol. 19, Num. 3, 1993.
- [Hobbs 1978a] Hobbs, J.R. *Coherence and Coreference*. SRI International. Technical Note 168, August 1978.
- [Hobbs 1978b] Hobbs, J.R. *Why Is Discourse Coherent?* SRI International. Technical Note 176, November 1978.
- [Hobbs 1978c] Hobbs, J.R. *Resolving Pronoun References*. *Lingua* 44 (1978):11-338.
- [Hobbs 1985] Hobbs, J.R. *On the Coherence and Structure of Discourse*. Technical Report CLSI-85-37, Center for the Study of Language and Information, Stanford, CA, October 1985.

- [Hovy 1990] Hovy, E.H. *Parsimonious and Profligate Approaches to the Question of Discourse Structure Relations*. In Proceedings of the 5th International Workshop on NLG, Pittsburgh, 1990.
- [Hovy 1993] Hovy, E. *Automated Discourse Generation using Discourse Structure Relation*. Artificial Intelligence, Vol. 63, Num. 1-2, October 1993.
- [Hutchins 1977] Hutchins, W.J. *On the Problem of Aboutness in Document Analysis*. Journal of Informatics 1(1):17-35, 1977.
- [Hutchins 1984] Hutchins, W.J. *Machine Translation and Machine-Aided Translation*. In Translation: Literary, Linguistic, and Philosophical Perspectives. William Frawley (Ed.). University of Delaware Press, 1984.
- [Hutchins 1985] Hutchins, W.J. *Information Retrieval and Text Analysis*. In New Approaches to the Analysis of Mass, Media, Discourse and Communication. T.A. van Dijk (Ed.), Gruyter, Berlin, 1985.
- [Hutchins 1987] Hutchins, W.J. *Summarization: Some Problems and Methods*. Meaning: The Frontier of Informatics. K. Jones (Ed.), Cambridge, London, 1987.
- [Jordan 1991] Jordan, M.P. *The Linguistic Genre of Abstracts*. In A. Della Volpe (ed.), The Seventeenth LACUS Forum. Linguistics Association of Canada and the United States, 1991.
- [Koch 1989] Koch, I.V. *A Coesão Textual*. Editora Contexto, 1989.
- [Koch e Travaglia 1990] Koch, I.V. e Travaglia, L.C. *A Coerência Textual*. Editora Contexto, 1990.
- [Krulee 1991] Krulee, G.K. *Computer Processing of Natural Language*. Prentice Hall International, Inc., 1991.
- [Lewis 1992] Lewis, D. *Computers and Translation*. In Computers and Written Texts. Christopher S. Butler. Blackwell, 1992.
- [Litman 1994] Litman, D.J. *Classifying Cue Phrases in Text and Speech using Machine Learning*. Proceedings of the Twelfth National Conference on Artificial Intelligence, Vol. 1, July 31 - August 4, 1994, Seattle, Washington.
- [Lobato 1986] Lobato, M.L.P. *Sintaxe Gerativa do Português. Da Teoria Padrão à Teoria da Regência e Ligação*. Vigília, 1986.
- [Lopes Fávero 1991] Lopes Fávero, L. *Coesão e Coerência Textuais*. Editora Ática, 1991.
- [Mann e Thompson 1983] Mann, W.C. e Thompson S.A. *Relational Propositions in Discourse*. Information Sciences Institute, Technical Report RR-83-115, November 1983.

- [Mann e Thompson 1987] Mann, W.C. e Thompson S.A. *Rhetorical Structure Theory: A Theory of Text Organization*. ISI Reprint Series, ISI/RS-87-190, June 1987.
- [Mateus et al. 1983] Mateus, M.H.M., Brito, A.M., Duarte, I.S. y Faria, I.H. *Gramática da Língua Portuguesa*. Livraria Almedina, Coimbra, 1983.
- [McCord 1986] McCord, M. *Natural Language Processing and Prolog*. Research Report. RC 11973 53899, IBM Thomas J. Watson Research Center, 1986.
- [McCord 1989] McCord, M. *Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars*. Lecture Notes in Natural Language and Logic, Number 459, Springer-Verlag 1989.
- [McKeown 1985] McKeown, K.R. *Discourse Strategies for Generating Natural-Language Text*. Artificial Intelligence, Vol. 27, 1985.
- [Milanez 1987] Milanez, L.F. *Considerações sobre Programas de Avaliação de Docentes pelos Alunos*. Rev. Ensino Eng., 6(1):47-52, São Paulo, 1º Sem. 1987.
- [Moore e Paris 1994] Moore, J.D. e Paris, C.L. *Planning Text for Advisory Dialogues: Capturing Intentional and Rhetorical Information*. Computational Linguistics, Vol. 19, Num. 4, 1994.
- [Morris e Hirst 1991] Morris, J. e Hirst, G. *Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of the Text*. Computational Linguistics, Vol. 17, Num. 1, 1991.
- [Oller e Giudici 1989] Oller, C.A.N. e Giudici, R. *Um Experimento Didático para Determinação de Tempos de Residência em Reator de Leito Fixo*. Rev. Ensino Eng., São Paulo, 8(1):16-22, 1º sem. 1989.
- [Pansani 1971] Pansani, C. *Análise Sintática*. Gráfica Forte Ltda., 1971.
- [Pereira e Warren 1980] Pereira, F.C.N. e Warren, D.H.D. *Definite Clause Grammars for Language Analysis. A Survey of the Formalism and the Comparison with Augmented Transition Networks*. Artificial Intelligence, Vol. 13, 1980.
- [Pimenta Rodrigues e Pereira Lopes 1992] Pimenta Rodrigues, I. e Pereira Lopes, J.G. *Discourse Temporal Structure*. Proceedings of the COLING'92, 1992.
- [Quirk et al. 1985] Quirk, R., Greenbaum, S., Leech, G. e Svartvik, J. *A Comprehensive Grammar of the English Language*. Longman, Londres, 1985.
- [Raposo 1992] Raposo, E.P. *Teoria da Gramática. A Faculdade da Linguagem*. Ed. Caminho, Lisboa, 1992.
- [Rodrigues 1988] Rodrigues, G.S. *Impactos Extensivos e Intensivos, sua Previsão, Avaliação e Medidas de Proteção*. Rev. Ensino Eng., São Paulo, 7(1):19-25, 1º sem. 1988.

- [Saggion e Carvalho 1994] Saggion, H. e Carvalho, A. *Anaphora Resolution in a Machine Translation System*. International Conference Machine Translation: Ten Years On. Cranfield, Inglaterra, 12 ao 14 de novembro 1994.
- [Saggion e Carvalho 1995a] Saggion, H. e Carvalho, A. *Análise Textual Visando a Tradução Automática*. XXII Congresso da Sociedade Brasileira de Computação e XXI Panel Latino Americano de Informática, Canela-RS, Brasil, 31 de julho ao 4 de agosto de 1995.
- [Saggion e Carvalho 1995b] Saggion, H. e Carvalho, A. *Text Structure Aiming at Machine Translation*. Reunión Nacional de Inteligencia Artificial 1995, Cuernavaca, Mexico, 20 ao 23 de setembro de 1995.
- [Saggion e Carvalho 1995c] Saggion, H. e Carvalho, A. *Cálculo de la Estructura de un Texto en un Sistema de Procesamiento de Lenguaje Natural*. XI Congreso de Lenguajes Naturales y Lenguajes Formales, 25 ao 29 de setembro de 1995, Tortosa, España.
- [Schank e Abelson 1977] Schank, R. e Abelson, R. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Publishers, 1977.
- [Sidner 1978] Sidner, L.S. *The Use of Focus as a Tool for Disambiguation of Definite Noun Phrases*. TINLAP-2, 1978.
- [Siegel e McKeown 1994] Siegel, E.S. e McKeown, K.R. *Emergent Linguistics Rules from Inducing Decision Trees: Disambiguating Discourse Clue Words*. Proceedings of the Twelfth National Conference on Artificial Intelligence, Vol. 1, July 31 - August 4, 1994, Seattle, Washington.
- [Sperandio 1987] Sperandio, C.A. *A Falha da Segurança do Trabalho está no Ensino?* Rev. Ensino Eng., São Paulo, 6(1):76-84, 1^o sem. de 1987.
- [Toropy e Aita 1987] Toropy, B. de O. e Aita, T.J. *O Ensino de Desenho no 1^o e 2^o graus e seus Reflexos no Ensino Superior*. Rev. Ensino Eng., São Paulo, 5(2):210-222, 2^o sem. 1986.
- [Weil et al. 1963a] Weil, B.H., Zarembek, I. y Owen, H. *Technical-Abstracting Fundamentals. Introduction*. Journal of Chemical Documentation, Num 3, 1963.
- [Weil et al. 1963b] Weil, B.H., Zarembek, I. y Owen, H. *Technical-Abstracting Fundamentals. Writing Principles and Practices*. Journal of Chemical Documentation, Num 3, 1963.
- [Weissberg e Buker 1990] Weissberg, R. e S. Buker. *Writing UP Research*. Prentice-Hall, Inc., 1990.
- [Weyne 1987] Weyne, de S.G.R. *O Ensino de Projetos de Indústrias Químicas no University College de Londres, Inglaterra, Um Estudo Comparativo com as*

- Condições Brasileiras*. Rev. Ensino de Eng., São Paulo, 6(1):89-96, 1^o sem. de 1987.
- [Wilks 1973a] Wilks, Y. *An Artificial Intelligence Approach to Machine Translation*. In *Computer Models of Thought and Language*, Schank, R. e Colby, K. (Eds.), Freeman, San Francisco, 1973
- [Wilks 1973b] Wilks, Y. *Preference Semantics*. Report Number CS-377, Stanford Artificial Intelligence Laboratory, 1973.
- [Winograd 1973] Winograd, T. *A Procedural Model of Language Understanding*. *Computer Models of Thought and Language*, R. Schank, K. Colby (Eds.), 1973.