

# INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

## PARTE 7. PROCESSAMENTO DE LINGUAGEM NATURAL

### 7.1. Introdução

O processamento de linguagem natural pode ser definido de formas diferentes. Todas as definições incorporam a noção de armazenamento em computador e manipulação de dados lingüísticos. Em outras palavras, o processamento de linguagem natural pode ser definido como a habilidade de um computador em processar a mesma linguagem que os humanos usam no dia-a-dia.

O processamento de linguagem natural é geralmente dividido em seis grandes áreas (Obermeier, 1987): (1) interfaces em linguagem natural para base de dados, (2) tradução de máquina - isto é, de uma linguagem natural para outra, (3) perscrutação de texto / programas de indexação inteligentes para sumarização de grandes quantidades de texto, (4) geração de texto para produção automática de documentos padrões, (5) sistemas de fala para permitir interação de voz com computadores, e (6) ferramentas para desenvolver sistemas de processamento de linguagem natural para aplicações específicas.

### 7.2. Abordagens do processamento de linguagem natural

O problema central dos sistemas de processamento de linguagem natural é a transformação de uma frase de entrada potencialmente ambígua em uma forma não ambígua que possa ser usada internamente por um sistema de computador. Estas representações internas variam, é claro, de uma aplicação para outra.

A transposição de uma frase potencialmente ambígua para uma representação interna é conhecida como "parsing" (análise). A palavra "parse" é derivada do latim: *pars orationis* (parte do discurso). No processamento de linguagem natural, *parsing* é usualmente um processo de combinar os símbolos de uma frase em um grupo que pode ser substituído por um outro símbolo mais geral. Este novo símbolo pode por sua vez ser combinado em um outro grupo, e assim por diante, até que uma estrutura permitida apareça.

Cinco tipos diferentes de *parsers* (analísadores) existem: casamento de padrões, baseado em gramática, semântico, baseado em conhecimento, e analisadores por redes neurais. Cada um tem uma abordagem única, sem igual, em processamento de linguagem natural.

*Abordagens por Casamento de Padrões.* Os primeiros programas de linguagem natural eram baseados na idéia de que os analisadores podiam procurar padrões lingüísticos numa sentença sem usar qualquer formalismo gramatical explícito. Durante a análise da sentença, o sistema apenas

procura por um possível casamento com um número fixo de padrões. Se um casamento é encontrado, o sistema executa determinada ação (por exemplo, reorganizar a entrada de acordo com um outro padrão). Ex.: ELIZA, o programa de casamento de padrões mais conhecido, foi projetado para simular um psicólogo Rogeriano. O programa foi escrito por Joseph Weizenbaum em 1966 e consiste de um conjunto de padrões, onde cada padrão tem um número de respostas associadas. Quando um determinado padrão é encontrado, o programa seleciona uma resposta do conjunto e faz todas as substituições necessárias nesta resposta. Um pouco do diálogo com ELIZA (Rich & Knight, 1994):

Pessoa: Homens são todos iguais.  
 ELIZA: *De que forma?*  
 Pessoa: Eles sempre estão nos aborrecendo com uma coisa ou outra.  
 ELIZA: *Você pode dar um exemplo específico?*  
 Pessoa: Bem, meu namorado me fez vir aqui.  
 ELIZA: *Seu namorado a fez vir aqui.*  
 Pessoa: Ele diz que eu ando deprimida a maior parte do tempo.  
 ELIZA: *Sinto ouvi-la dizer que está deprimida.*  
 Pessoa: É verdade. Estou infeliz.  
 ELIZA: *Você acha que vir aqui ajudará você se sentir menos infeliz?*  
 Pessoa: Eu preciso de ajuda, isto parece certo.  
 ELIZA: *O que significa para você se você conseguir ajuda?*  
 Pessoa: Talvez eu pudesse aprender a conviver melhor com minha mãe.  
 ELIZA: *Fale-me mais sobre sua família.*

Os programas por casamento de padrões sem uma base gramatical provaram ser de uso limitado. Eles são úteis apenas se se necessita de análise parcial ou se outros componentes do sistema podem compensar a perda de informação sintática. O conceito básico de casamento de padrões, entretanto, foi usado nas gramáticas semânticas.

*Abordagens baseadas em Gramática.* O termo gramática se refere a um conjunto de regras que descrevem quais sentenças são parte de uma determinada linguagem. Estas regras são freqüentemente chamadas de regras de rescrita ou produções. Duas regras simples de rescrita são:

$$S \rightarrow SN \text{ } SV$$

$$S \rightarrow SV$$

Estas regras estipulam que uma sentença (S) deve ter um sintagma nominal (SN) e um sintagma verbal (SV), ou apenas um sintagma verbal. Com estas regras pode-se construir uma estrutura em árvore mostrando sem ambigüidade como as palavras interagem numa sentença (veja figura 1). As folhas da árvore são chamadas de palavras terminais. Os outros símbolos, em maiúscula, são não-terminais. As regras de rescrita usualmente estabelecem como um não-terminal pode ser rescrito como uma cadeia de terminais ou outros não-terminais.

Gramática:

$S \rightarrow SN\ SV$   
 $SN \rightarrow DET\ SN1$   
 $SN \rightarrow SN1$   
 $SN1 \rightarrow ADJ\ SN1$   
 $SN1 \rightarrow SUBST$   
 $SV \rightarrow VERBO\ ADV$   
 $SUBST \rightarrow programa$   
 $VERBO \rightarrow compila$   
 $ADJ \rightarrow novo$   
 $DET \rightarrow o$   
 $ADV \rightarrow vagarosamente$

Sentença:

“o novo programa compila vagarosamente”

Árvore:

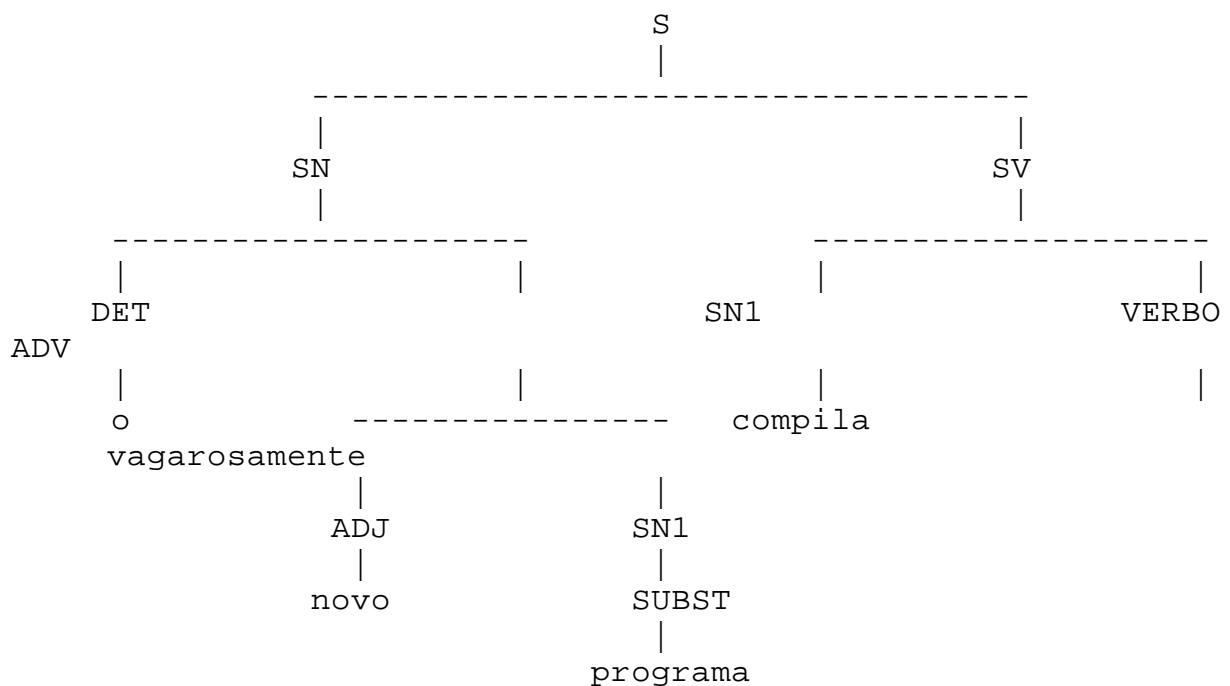


Figura 1

O lingüista Noam Chomsky dividiu a gramática em 4 tipos, baseado nos tipos de regras que elas usavam. A gramática mais simples, a de tipo 3, é conhecida como *gramática de estado finito* ou *regular*. Ela pode produzir apenas sentenças simples.

A gramática do tipo 2 é conhecida como *gramática livre de contexto*. Nesta gramática, o lado esquerdo de cada regra de rescrita pode consistir de apenas um único símbolo não terminal. Note que este símbolo sempre pode ser rescrito como o lado direito da regra, não importando o contexto no qual esse não terminal aparece.

A outra gramática complexa é a do tipo 1 ou gramática *sensível ao contexto*. Neste tipo, mais de um símbolo pode aparecer no lado esquerdo da regra de rescrita. Existe apenas um requisito para estas regras: deve existir mais símbolos do lado direito que do esquerdo.

A gramática mais complexa, do tipo 0, tem regras que não seguem nenhum padrão. Esta gramática é muito difícil de analisar. Realmente, os pesquisadores têm provado que uma *Máquina de Turing* é necessária para processar linguagens com esta gramática.

Chomsky argumentava que uma linguagem natural como o Inglês, e por conseqüência, o Português, não poderia ser completamente descrita por uma gramática livre de contexto, mas uma sensível ao contexto ou do tipo 0.

Um argumento favorável ao Português ser uma linguagem sensível ao contexto envolve o fato que substantivos singulares e plurais requerem verbos singulares e plurais, respectivamente. Por exemplo, a gramática da figura 1 é uma gramática simples sensível ao contexto. Pode-se expandir esta gramática adicionando três regras de rescrita, todas permitidas:

SUBST → programas

VERBO → compilam

DET → os

Agora, portanto, a gramática permitirá uma sentença que não é permitida no Português:

“Os programas compila vagorosamente”.

*Abordagens semânticas.* O papel central do significado no entendimento da linguagem tem levado os pesquisadores a considerar mais a abordagem semântica do que a sintática. Estes pesquisadores não negam a necessidade de algum processamento estrutural; eles usam a análise sintática para complementar suas considerações semânticas.

Duas abordagens orientadas semanticamente em processamento de linguagem natural são a gramática de caso e a gramática semântica.

A idéia da *gramática de caso* é que toda sentença tem uma representação "implícita" de seu significado. Esta representação inclui o verbo e os vários sintagmas nominais relacionados com o verbo. Por exemplo, na sentença "João abriu a porta com uma chave" designamos "João" como o agente, "porta" como o objeto, e "chave" como o instrumento para o verbo "abriu". Note que os casos permanecem os mesmos para a sentença "A porta foi aberta por João com uma chave".

O descobridor da teoria de caso, C. Fillmore, se referiu aos relacionamentos entre esses substantivos e os verbos como casos. Fillmore estabeleceu vários casos, nenhum deles ocorrendo mais de uma vez em uma sentença.

A *gramática semântica* consiste de um léxico e uma série de regras de rescrita. É similar a uma gramática sintática, exceto que classes de palavras (ex. SUBSTANTIVO, VERBO) são substituídas por classes semânticas específicas (ex. NAVIOS, PROPRIEDADES-NAVIOS). A vantagem desta abordagem é que o tamanho destas classes semânticas é muito menor do que o tamanho de uma classe de palavra equivalente. Isto resulta em uma estratégia de análise muito mais eficiente, desde que o programa tem que checar um número menor de possibilidades. A desvantagem da gramática semântica é a dificuldade de transferir regras de rescrita de um domínio de aplicação para outro.

*Abordagens baseadas em conhecimento.* Ao invés de contar apenas com a informação semântica ou estrutural de uma sentença, alguns sistemas de processamento de linguagem natural também têm acesso a uma base de conhecimento para um domínio específico do conhecimento. Isto contrasta com a maioria das teorias baseadas em gramática que olham o processamento de linguagem natural simplesmente em termos de um conjunto de regras de rescrita para o processamento a nível da sentença.

Uma abordagem baseada em conhecimento é chamada de *análise especialista em palavra*. Nesta abordagem, a palavra é considerada a unidade lingüística básica. O conhecimento lingüístico é distribuído entre um grupo de "especialistas" procedurais que sabem como a interpretação de uma palavra muda em determinados contextos.

Um argumento em favor desta abordagem é o fato de que as palavras têm uma estrutura lingüística e conceptual ricas. Também é pouco provável que a linguagem possa ser reduzida somente a varias regras de rescrita, como tratam muitas teorias baseadas em gramática.

Uma outra abordagem baseada em conhecimento é chamada de *teoria da dependência conceptual*. A idéia central por trás desta teoria é criar uma representação canônica de uma sentença, baseado em certas primitivas semânticas. Uma representação canônica é simplesmente uma forma básica de representar o significado de uma sentença. Sentenças diferentes que signifiquem a mesma coisa terão a mesma representação canônica. Por exemplo, "Jean come doce" e "o doce foi comido por Jean" ambas têm a mesma representação canônica: "Jean  $\leftrightarrow$  INGEST  $\leftrightarrow$  doce."

Nesta teoria, as primitivas semânticas são as entidades mais básicas usadas para descrever o mundo. Palavras individuais podem sempre ser analisadas de novo, mas primitivas semânticas não.

*Abordagem por rede neural.* Uma abordagem mais recente em processamento de linguagem natural envolve estabelecer uma rede de unidades de computação parecidas com o neurônio. Cada unidade tem várias entradas, um conjunto pequeno de estados possíveis, e uma saída que é uma função das entradas. Cada entrada para a unidade de computação tem um peso de

conexão (peso sináptico), que pode variar de -1 a 1. Quando uma unidade é ativada, ela analisa todas as suas entradas e as pondera de acordo com seus respectivos pesos de conexão. Se certas condições são encontradas, a unidade gera um valor de saída que é usado como entrada por outras unidades. Note que apenas os pesos de conexão das entradas podem ser mudados durante o "aprendizado"; o padrão de conexão é estabelecido previamente.

Este tipo de sistema é usualmente chamado de abordagem por rede neural ou *conexionista*. A premissa fundamental desta abordagem é que as unidades individuais não transmitem grandes quantidades de dados, mas o processamento ocorre porque grande número de unidades similares são conectadas entre si.

Um modelo de análise por rede neural contém três níveis de "neurônios". O primeiro nível é o nível lexical, que serve como nível de entrada da rede. Aqui, os neurônios são mapeados em determinadas palavras. No segundo nível, o nível do sentido da palavra, as entradas do nível lexical são combinadas para ativar neurônios que representam o significado das palavras. No terceiro nível, de lógica de caso, os significados são combinados para formar predicados e objetos.

A análise por rede neural se aproxima do modelo do processamento de informação lingüística humano, baseado na evidência neurológica.

### 7.3. Níveis de análise da linguagem

Os dados estruturais a serem identificados e analisados correspondem aos níveis de análise da linguagem. Como a maioria dos programas tratam apenas de material escrito, o nível fonológico (sistema sonoro) é deixado de lado. Cinco níveis podem ser descritos, exemplificados pela seguinte descrição de processamento da sentença simples "O sistema recuperou os documentos" (Warner, 1988).

1. *Morfológico* - palavras (conjunto de letras separadas por espaço) são decompostas em raízes e terminações. Por exemplo, o termo "documentos" seria quebrado na raiz "documento" e na terminação (plural) "s".

2. *Lexical* - usando um dicionário, a cada raiz é atribuído um conjunto de categorias léxicas. Por exemplo, ao termo "documento" deve ser atribuído a categoria lexical SUBSTANTIVO.

3. *Sintático* - usando um módulo de programa chamado de analisador sintático, uma estrutura gramatical é atribuída à sentença. O programa pega o nível de entrada da palavra do componente lexical e decide como as palavras individuais se ligam para formar sintagmas, cláusulas e sentenças inteiras. A seguinte análise da sentença resultaria:

	SN	-----	(Artigo)	O
/		\-----	(Substantivo)	sistema
S		/-----	(Verbo)	recuperou

\ SV - SN -(Artigo) os  
                  \ (Substantivo) documentos

4. *Semântico* - a estrutura sintática é traduzida para alguma forma lógica que representa o significado da sentença e permitirá se fazer certas inferências. Por exemplo, dada uma representação semântica apropriada, a questão "Os documentos estão no sistema?" poderia ser respondida - o sistema poderia inferir que a recuperação de documentos significa que havia documentos a serem recuperados.

5. *Pragmático* - este analisa a sentença no contexto, levando em consideração um certo corpo de conhecimento sobre o domínio e sobre os planos e metas do falador e do ouvitor na conversação. Por exemplo, a informação pragmática permitiria ao sistema inferir que um computador foi envolvido na operação de recuperação, ainda que isto não esteja explícito na sentença.

Os sistemas de processamento de linguagem natural diferem nas quantidades dos cinco tipos de conhecimento os quais eles podem explorar e em como o conhecimento é configurado em uma arquitetura de sistemas global.

#### 7.4. Técnicas de análise

Existem dois grupos de técnicas de análise: não-determinística e determinística (Obermeier, 1987). Os não-determinísticos podem ser divididos em analisadores *top-down* e *bottom-up*.

Os analisadores *top-down* tentam casar as regras da gramática com a entrada, começando na regra de rescrita mais alta (que usualmente envolve o símbolo inicial ou símbolo da sentença S) e recursivamente se move em direção à mais baixa, a regra de rescrita mais específica. O analisador é bem sucedido se uma sentença pode ser construída a partir da sentença de entrada (casando).

Os analisadores *top-down* são fáceis de escrever e de modificar. Regras que são mais usadas podem ser colocadas na frente de regras menos usadas, aumentando a performance. E o número das sentenças geradas pode ser limitada arbitrariamente.

Contudo, os analisadores *top-down* podem ser lentos. Se todas as regras de um nível falham, o analisador volta ("backtracks") para o nível anterior para tentar uma outra regra lá. Durante o *backtracking*, os mesmos componentes podem ser analisados muitas vezes. Os analisadores *top-down* também têm problemas para manipulação de entrada disforme e requerem um módulo separado para decidir qual dos muitos analisadores bem sucedidos é o melhor.

Os analisadores *bottom-up* começam combinando os elementos do nível mais baixo primeiro e então constróem para cima componentes maiores. Por exemplo, na figura 1, os primeiros passos de um analisador *bottom-up* seriam: substituir o não terminal DET por "o"; ADJ por "novo"; SUBST por "programa"; e SN1 por ADJ SUBST.

Os analisadores *bottom-up* podem, ao menos parcialmente, analisar entrada disforme. Também, mecanismos podem ser aplicados para reduzir a explosão combinatorial de possíveis



análises. Como os analisadores *bottom-up* não são direcionados à meta, eles geram numerosas análises espúrias. E a correção da análise só pode ser determinada depois que todas as análises forem realizadas.

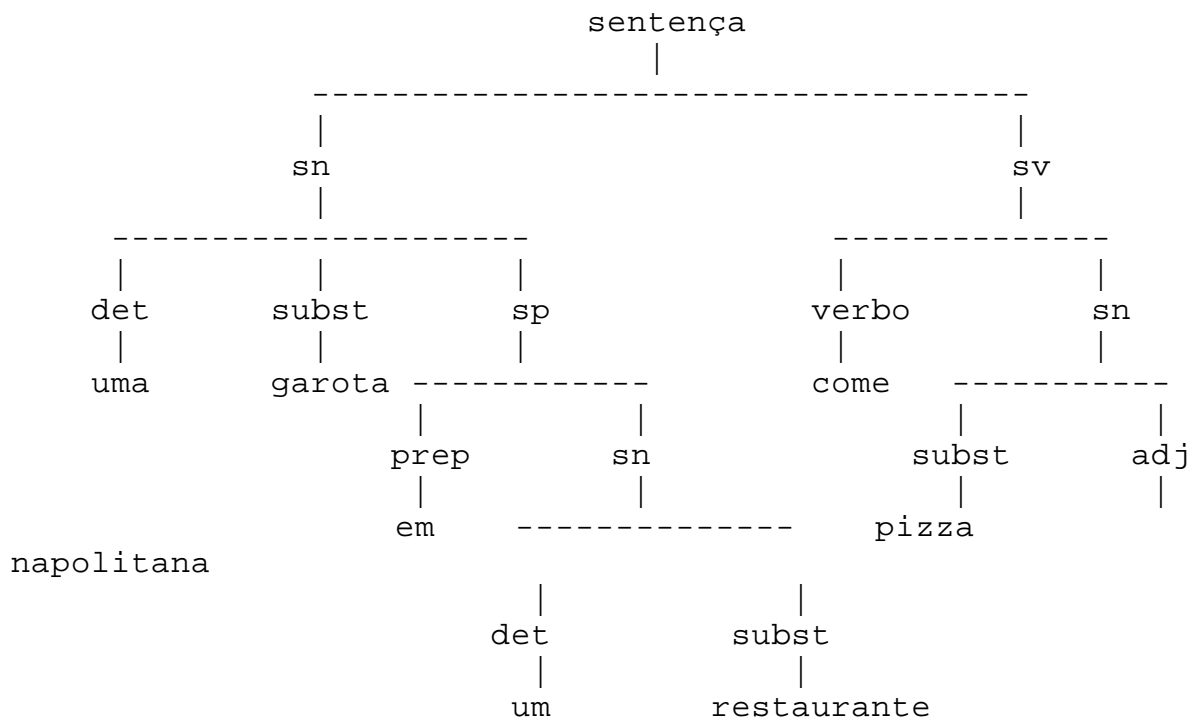
A análise *determinística* não tem *backtracking*. É também chamada de análise "espere-e-veja". A técnica cria novos nós de um modo *bottom-up* mas usa uma característica limitada de "olhar para frente" para determinar qual nó usar. Uma vantagem do determinístico é o aumento de velocidade porque ele evita a explosão combinatorial para possíveis análises. A desvantagem é que o algoritmo é baseado somente na informação sintática.

## 7.5. Interpretação da linguagem

Os principais aspectos na habilidade de um computador em entender uma sentença em linguagem natural são a análise sintática, a adição de características às palavras, a alocação de funções às palavras, e a fixação da sentença na sua estrutura de contexto. A adição de características novas às palavras significa a incorporação no sistema de entendimento iterativo características como número e gênero. A alocação de funções às palavras significa explicitamente a expressão das funções executadas por cada palavra na sentença; estas funções incluem o fato da palavra ser um agente ou um objeto. As pessoas usam estas mesmas técnicas mas não necessariamente na mesma ordem.

As pessoas e os computadores devem trabalhar com a sintaxe e a semântica cooperativamente, mas existem muitas formas nas quais sintaxe e semântica podem interagir. A análise pode produzir uma representação sintática completa que é então interpretada semanticamente. Usar uma fase de análise separada para representar concisamente os relacionamentos sintáticos é uma vantagem porque muitas regras semânticas diferentes pedem essencialmente a mesma informação sintática.

*Análise Sintática.* Um analisador sintático aplica regras de estrutura de frase para construir uma árvore de análise. Dados um conjunto de regras e um léxico apropriado, pode-se construir uma árvore de análise, que é uma estrutura sintática que um sistema de entendimento semântico pode usar. O propósito da análise sintática é checar a validade da sentença de entrada e organizar a sentença em unidades sintáticas que determinam seu significado. A figura abaixo mostra a árvore de análise sintática da sentença "A garota em um restaurante come pizza napolitana".



Usando os fatos e as cláusulas de Horn do Prolog, o analisador pode construir a estrutura lista:

sentença (sn (det (uma), subst (garota), sp (prep (em), det (um), subst (restaurante))), sv (verbo (come), sn (subst (pizza), adj (napolitana)))).

*A adição de características às palavras.* Para checar a compatibilidade de número e gênero, consistência de tipos de verbos, e certas inconsistências semânticas, deve-se adicionar informação ao analisador sobre palavras definindo suas características. Com esta informação, o analisador pode fazer um cheque de consistência para eliminar estruturas incorretas em sintaxe e em semântica (Geetha e Subramanian, 1990):

. Compatibilidade de número. Considere a sentença "Ele gostam de comida quente". O número do substantivo e do verbo não casam e então a sentença está sintaticamente incorreta. A incompatibilidade está entre palavras pertencentes a frases sintáticas diferentes, que são os grupos formados por palavras da sentença baseados nas suas categorias sintáticas (como sintagma nominal e sintagma verbal).

Considere um outro exemplo: "Maria comeu estas manga". Neste caso, a incompatibilidade está entre palavras pertencentes à mesma frase sintática. Isto traz a necessidade de adicionar o número como uma característica da palavra.

. Compatibilidade de gênero. Considere a sentença "João e o amigo dela foram ao parque". Quando considerada isolada, esta sentença mostra que os gêneros de "João" e "dela" não casam, então a

conclusão é que o amigo não é de João. Para achar a incompatibilidade, é necessário conhecer os gêneros de "João" e "dela". Então, deve-se adicionar o gênero como uma característica da palavra.

. Consistência de tipo de verbos. Verbos têm muitas características que ditam a estrutura do resto da sentença. Considere a sentença "Henry dormiu Tom". "Dormir" é um verbo intransitivo, que significa que ele não age num objeto específico, então ele não pode ter um SN como objeto direto. No máximo, sintagmas preposicionais ou adverbiais como "com Tom" ou "ruidosamente". Portanto, a analisador deve ter a informação se o verbo é transitivo ou intransitivo.

. Inconsistências semânticas. Número, gênero e tipo de verbo são usados apenas para checar o aspecto sintático das sentenças. Pode-se usar certas características de palavras para achar as inconsistências semânticas que dependem somente da natureza de determinadas palavras. Considere a sentença "Ivan come pessoas". O analisador checa as características do sujeito ("Ivan") e objeto ("pessoas") para ver se eles são semanticamente compatíveis com o verbo ("come"). Neste exemplo, o verbo prefere um sujeito animado e um objeto não humano, então existe uma inconsistência semântica mesmo que a sentença esteja sintaticamente correta. (Para as exceções, como Ivan ser um canibal, adicionar-se-ia este fato ao sistema, que daria precedência sobre as regras gerais no léxico do sistema.)

*A estrutura lexical.* O léxico que o analisador usa para sua análise em linguagem natural deve ter dois tipos de informação: a informação sintática e os detalhes semânticos. Ele usa esta informação para auxiliar a análise sintática e mais tarde construir a representação contextual e semântica. A forma da entrada lexical é diferente para categorias sintáticas diferentes.

Substantivos, adjetivos, advérbios, e outras partes do discurso - exceto verbos - têm a mesma estrutura no léxico:

tipo\_sintático da palavra (X, R, N, G, T)

O tipo\_sintático da palavra seria um *token* como *é\_substantivo*, *é\_pronome*, ou *é\_nome\_próprio*. X é a palavra do texto de entrada, R é a raiz da palavra, N é o número da palavra (singular, plural, ou indeterminado), G é o gênero (masculino, feminino, comum, ou neutro) e T é uma lista contendo propriedades semânticas como humano, animado, ou entidade.

Para verbos, o léxico também inclui características de verbo como tipo (transitivo ou intransitivo) para análise semântica; estas características podem ser incluídas como regras separadas. Pode-se usar outras características (como número e tempo verbal) diretamente (por casamento de padrões simples) para achar anomalias sintáticas como discordância de sujeito-verbo (por exemplo, "Ele comem"). A estrutura para verbos do léxico é:

é\_verbo (X, R, N, TN, T)

onde X é a palavra do texto de entrada, R é a raiz do verbo, N é o número, TN é o tempo verbal (como passado simples, presente, futuro, etc.) e T é o tipo (transitivo ou intransitivo). Para a língua portuguesa, haveria também a necessidade da pessoa.

Depois da análise sintática, o analisador agrupa as palavras em frases sintáticas como SN, SV e SP. Cada frase sintática tem uma palavra principal da qual a frase depende, e a frase sintática aceita as características desta palavra principal. O sistema determina quais funções estas frases executam, aplicando regras apropriadas e casamento de padrões.

*A representação semântica.* As funções reais, as quais definem as relações que as outras frases sintáticas têm com o verbo, são executadas por frases sintáticas operando como uma unidade. O conhecimento das diferentes funções executadas por estas frases dá ao computador uma compreensão parcial do significado da sentença. O significado da estrutura inteira depende do verbo e das outras unidades que executam funções relacionadas com o verbo principal. Cada verbo terá um *frame*, o qual será armazenado no léxico. Outros constituintes sintáticos serão alocados aos seus respectivos *slots* se eles satisfizerem os limites semânticos dados.

## 7.6. Exercícios

1. Escreva um programa Prolog, para fazer o "parsing" de sentenças como:

"José ama Maria".  
"Todo homem alto caiu".

Construa um léxico apenas contendo as palavras necessárias para analisar estas duas sentenças. Use Gramática de Cláusulas Definidas. Inclua concordância de gênero e número.

## BIBLIOGRAFIA

- Geetha, T. V., Subramanian, R. K. (1990). "Representing Natural Language with Prolog". *IEEE Software*. pages 85-92, vol. 7, n. 2.
- McClelland, J. L., Rumelhart D. E. and the PDP Research Group. (1986). Parallel Distributed Processing - Explorations in the Microstructure of Cognition. Volume 2: Psychological and Biological Models. A Bradford Book, The MIT Press.
- Obermeier, K. K. (1987). "Natural-Language Processing". *Byte*, pages 225-232. December.
- Rich, E. & Knight, K. (1994). Inteligência Artificial. 2ª. edição. Makron Books.
- Rosa, J. L. G. & Netto, M. L. A. (1993). "Processamento de Linguagem Natural - Uma abordagem conexionista". *Revista de Informática*, no. 1, pp. 9-15, Junho - Instituto de Informática - PUCCAMP.
- Schwind, C. B. (1985). "Logic Based Natural Language Processing". Natural Language Understanding and Logic Programming. V. Dahl and P. Saint-Dizier (Editors) pages 207-219. Elsevier Science Publishers B.V. (North-Holland).

- Warner, A. J. (1988). "Natural Language Processing in Information Retrieval". *Bulletin of the American Society for Information Science* - pages 18-19. August/September.