

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
Disciplina: Arquiteturas Especiais de Computadores
Professor Dr. Philippe Olivier Alexandre Navaux

Processamento de linguagem natural: princípios básicos e a implementação de um *analisador sintático* de sentenças da língua portuguesa

por

[FABIO ABREU DIAS DE OLIVEIRA](#)

Sumário

Resumo

1. Introdução

2. Processamento de linguagem natural

[2.1. Histórico](#)

[2.2. Etapas do processo](#)

[2.2.1. Análise morfológica](#)

[2.2.2. Análise sintática](#)

[2.2.3. Análise semântica](#)

[2.2.4. Pragmática](#)

[2.3. Exemplo de processamento](#)

3. Implementação de um analisador sintático de sentenças da língua portuguesa

[3.1. Processamento simbólico da língua natural](#)

[3.2. Prolog como linguagem para processamento de línguas naturais](#)

[3.3. Implementação](#)

4. Conclusões

Bibliografia Consultada

Resumo

Este trabalho apresenta as principais questões que devem ser resolvidas quando da implementação de sistemas processadores de linguagens naturais. No intuito de pôr em prática o estudo realizado, implementou-se um analisador sintático de sentenças escritas em português. O analisador recebe na entrada uma sentença e, como saída, indica se a sentença está sintaticamente correta ou não e, se estiver, mostra a árvore sintática da mesma. Tal analisador foi implementado em Prolog e baseou-se em um subconjunto da gramática

da língua portuguesa. Face às características da linguagem Prolog, o analisador aqui implementado pode ser facilmente estendido, agregando novas regras gramaticais e novas entradas no dicionário léxico.

[\[Voltar ao sumário\]](#)

1. Introdução

A tarefa de processar uma linguagem natural permite que os seres humanos comuniquem-se com os computadores da forma mais "natural" possível, utilizando a linguagem com a qual mais estão habituados. Elimina-se, desta maneira, a necessidade de adaptação a formas inusitadas de interação, ou mesmo o aprendizado de uma linguagem artificial, cuja sintaxe costuma ser de difícil aprendizado e domínio, a exemplo das *linguagens de consulta a bancos de dados*.

O processamento da linguagem natural emerge nos dias atuais como um instigante e desafiador campo de pesquisa. A motivação para tal estudo fundamenta-se no aperfeiçoamento da interação homem—máquina, a qual ainda é motivo de preocupação para muitos usuários, tendo em vista sua complexidade.

Os prognósticos para o futuro revelam a existência de máquinas capazes de compreender a língua natural, sendo capazes, por exemplo, de atender a comandos expressos na linguagem falada por usuários. Ademais, o desenvolvimento de modelos computacionais da língua natural permitirá um maior processamento de informações, visto que a maior parte do conhecimento humano está registrado na forma lingüística.

O caráter interdisciplinar do processamento de linguagem natural ilustra o nível de complexidade normalmente requerido às abordagens desenvolvidas, as quais se utilizam de conceitos de inteligência artificial, teoria da computação, compiladores, lingüística computacional e de outras disciplinas.

Basicamente, as aplicações dos sistemas que tratam a língua natural podem ser divididas em duas classes: aplicações baseadas em texto e aplicações baseadas em diálogos.

Exemplos de aplicações baseadas em texto são sistemas que procuram documentos específicos em uma base de dados (exemplo: encontrar livros relevantes em uma biblioteca), tradutores de documentos, e sistemas que resumem textos (exemplo: produzir 3 páginas resumidas de um livro de 100 páginas).

Com relação às aplicações baseadas em diálogos, pode-se citar as *interfaces de linguagem natural para bancos de dados*, os sistemas tutores e os sistemas que interpretam e respondem a comandos expressados em linguagem escrita ou falada.

Para um sistema ser considerado um tratador da língua natural, duas condições devem ser satisfeitas:

- um subconjunto de entrada e/ou saída do sistema é codificado em uma linguagem natural;
- o processamento da entrada e/ou a geração da saída é baseada no conhecimento sobre aspectos sintáticos, semânticos e/ou pragmáticos de uma linguagem natural.

Observa-se, principalmente com a segunda condição, a exigência da interpretação do conteúdo de uma sentença, sendo que esta condição não é satisfeita nos sistemas que processam a linguagem natural puramente como *strings* (por exemplo, editores de texto e pacotes estatísticos).

[\[Voltar ao sumário\]](#)

2. Processamento de linguagem natural

2.1. Histórico

No início da década de 50, começaram a surgir os primeiros trabalhos de pesquisa concernentes ao tratamento informatizado da língua natural.

O ano de 1968, com a publicação da obra de Arthur Clark – *2001 Odisséia no Espaço* – assistiu também ao aparecimento de um número de programas para a compreensão da linguagem natural. Nos anos 60, os computadores já eram capazes de aceitar e de responder a questões em inglês, com respeito a muitos assuntos (por exemplo, álgebra, medicina e relações de parentesco), e podiam conduzir uma entrevista psiquiátrica de nível rudimentar em inglês, galês ou alemão.

Quatro categorias históricas de programas em linguagem natural foram identificadas:

- alguns programas (por exemplo, BASEBALL, SAD-SAM, STUDENT e ELIZA) tinham por objetivo a geração de um número reduzido de resultados em domínios específicos. A simplicidade do processo permitia que muitos dos problemas da linguagem natural pudessem ser ignorados;
- em alguns dos primeiros sistemas, tais como o PROTO-SYNTHEX1, era armazenada uma representação do texto, recomendando-se a engenhos de indexação para auxiliar a recuperação de determinadas palavras ou frases. Como o texto armazenado podia cobrir qualquer assunto, os sistemas não eram restritos a um determinado domínio. Todavia, esses sistemas eram semanticamente fracos e não tinham poderes dedutivos;
- sistemas de lógica limitada (por exemplo, SIR, TLC, DEACON e CONVERSE) tinham por objetivo traduzir frases de entrada para a notação formal usada na base de dados. Aqui a intenção era permitir que se fizessem deduções a partir da informação mantida na base de dados, mesmo que somente alguns dos processos utilizados na conversação do dia-a-dia pudessem ser explorados;
- os sistemas com base de conhecimento (por exemplo, o LUNAR e o SHRDLU) usavam informação sobre um assunto específico para compreender as frases de entrada. Tais programas, alguns dos quais constituíam sistemas especialistas, exibiam vários poderes dedutivos.

Dentre os exemplos citados, destaca-se o programa ELIZA, desenvolvido por Joseph Weizenbaum, em 1966. O referido programa caracterizava-se por assumir o papel de um psiquiatra que conversava sobre os problemas de um paciente humano – o usuário. Deve-se mencionar, no entanto, que o ELIZA trabalhava apenas através de truques semânticos, não existindo qualquer compreensão do tema sobre o qual a conversa se desenvolvia.

[\[Voltar ao sumário\]](#)

2.2. Etapas do processo

Para que um sistema computacional interprete uma sentença em linguagem natural, é necessário manter informações morfológicas, sintáticas e semânticas, armazenadas em um dicionário, juntamente com as palavras que o sistema compreende. As etapas do processamento da linguagem natural estão discriminadas a seguir:

2.2.1. Análise morfológica

O analisador morfológico identifica palavras ou expressões isoladas em uma sentença, sendo este processo auxiliado por delimitadores (pontuação e espaços em branco). As palavras identificadas são classificadas de acordo com seu tipo de uso ou, em linguagem natural, categoria gramatical.

Neste contexto, uma instância de uma palavra em uma sentença gramaticalmente válida pode ser substituída por outra do mesmo tipo, configurando um sentença ainda válida (exemplo: substantivos, pronomes, verbos, etc.). Dentro de um mesmo tipo de palavra, existem grupos de regras que caracterizam o comportamento de um subconjunto de vocábulos da linguagem (exemplo: formação do plural de substantivos terminados em "ão", flexões dos verbos regulares terminados em "ar", etc.). Assim, a morfologia trata as palavras quanto a sua estrutura, forma, flexão e classificação, no que se refere a cada um dos tipos de palavras.

Um reconhecedor utilizado para a análise morfológica é o autômato finito. Foi proposta uma forma de comprimir vocabulários extensos através de autômatos determinísticos acíclicos minimizados.

O emprego do analisador morfológico é fundamental para a compreensão de uma frase, pois para formar uma estrutura coerente de uma sentença, é necessário compreender o significado de cada uma das palavras componentes [RIC 93].

[\[Voltar ao sumário\]](#)

2.2.2. Análise sintática

Através da gramática da linguagem a ser analisada e das informações do analisador morfológico, o analisador sintático procura construir árvores de derivação para cada sentença, mostrando como as palavras estão relacionadas entre si.

Durante a construção da árvore de derivação, é verificada a adequação das seqüências de palavras às regras de construção impostas pela linguagem, na composição de frases, períodos ou orações. Dentre estas regras, pode-se citar a concordância e a regência nominal e/ou verbal, bem como o posicionamento de termos na frase. Um termo corresponde a um elemento de informação (palavra ou expressão), e é tratado como unidade funcional da oração, participando da estrutura como um de seus constituintes, denominados sintagmas.

A análise sintática de uma oração em português deve levar em conta os seguintes sintagmas: termos essenciais (sujeito e predicado), termos integrantes (complementos verbal e nominal) e termos acessórios (adjunto adverbial, adjunto adnominal e aposto). A análise do período, por sua vez, deve considerar o tipo de período (simples ou composto), sua composição (por subordinação, por coordenação) e a classificação das orações (absoluta, principal, coordenada ou subordinada).

Nos sistemas de processamento de linguagem natural, o maior problema é a transformação de uma frase potencialmente ambígua em uma não ambígua, a qual será utilizada pelo sistema. Esta transformação é conhecida como *parsing*.

As abordagens de linguagens formais são utilizadas com muito sucesso no estudo da análise sintática. Dentre estas:

- **Gramáticas Regulares:** para o processamento sintático da linguagem natural, estas gramáticas são bastante simples e facilmente reconhecidas, porém, apresentam um poder de expressão limitado (equivalente ao poder de expressão de um autômato finito).
- **Gramáticas Livres de Contexto:** são muito úteis no que tange à descrição de gramáticas em linguagem natural. Em geral, são mais poderosas que as regulares, permitindo a representação de linguagens com um certo grau de complexidade. No entanto, a dificuldade em expressar dependências simples (exemplo: concordância entre verbo e sintagma nominal) constitui um dos maiores problemas para sua utilização no tratamento da língua natural. Abordagens puramente livres de contexto não são suficientemente poderosas para captar a descrição adequada deste gênero de linguagem. Ainda assim, é utilizada uma notação denominada como *Definite Clause Grammar* (DCG), disponível em Prolog, para definir gramáticas livres de contexto e analisar sentenças (realizar o *parsing*).
- **Gramáticas Sensíveis ao Contexto:** os problemas de dependência expressos anteriormente são resolvidos nesta classe de gramática. Ainda assim, as gramáticas sensíveis ao contexto não abordam satisfatoriamente o tratamento de restrições gramaticais [RIC 93]. O impedimento para sua utilização, contudo, reside na questão do reconhecimento. O problema de decidir se uma sentença pertence a uma gramática sensível ao contexto é uma função exponencial sobre o tamanho da sentença, o que torna a implementação do procedimento de verificação uma questão complexa, do ponto de vista computacional.

Cabe lembrar a existência das gramáticas irrestritas, também presentes na hierarquia de Chomsky, as quais não são utilizadas para a construção de interfaces em linguagem natural.

A maioria das pesquisas propõem, atualmente, trabalhar em modelos que se situem em um nível intermediário entre as gramáticas livres de contexto e as sensíveis ao contexto, aliando boa capacidade de representação, incluindo construções que permitam modelar dependências, e um modelo computacional viável.

[\[Voltar ao sumário\]](#)

2.2.3. Análise semântica

Significados compõem as estruturas criadas pelo analisador sintático. A questão da representação do significado apresenta diversas dificuldades. Pode-se mencionar a questão dos significados associados aos morfemas componentes de uma palavra (mercado, hipermercado), a questão da ambigüidade (tomar, em "tomar de alguém", em "tomar um banho" ou em "tomar suco"), ou a diferenciação entre significado e sentido ("casa", "minha casa").

O analisador semântico analisa o sentido das estruturas das palavras que foram reagrupadas pelo analisador sintático, uma vez que o analisador morfológico permitiu identificar estas palavras individualmente.

Pode-se dizer que a semântica se desdobra em semântica léxica e em semântica gramatical. A semântica léxica busca uma representação conceitual para descrever o sentido, sendo que, para construir esta representação, pode ser feita a decomposição semântica das unidades léxicas (em primitivas ou em traços semânticos), ou podem ser utilizadas redes semânticas. Esta última forma de representação foi originada da psicologia e leva em conta a forma como os seres humanos categorizam e memorizam conceitos.

A compreensão da relação entre as palavras é tão importante quanto a compreensão das próprias palavras. Enfoques formais para a semântica gramatical tentam descrever o sentido de uma frase mediante a tradução de sua estrutura sintática para uma fórmula lógica-semântica. Como não existe uma correspondência imediata e biunívoca entre sintaxe e semântica, uma mesma estrutura sintática pode dar origem a diferentes representações semânticas. Por exemplo, *uma professora de capoeira pernambucana* pode referir-se a uma pessoa nascida em Pernambuco, a qual ensina capoeira, ou a uma pessoa que ensina capoeira no estilo em que esta é praticada em Pernambuco.

Em suma, denomina-se análise semântica o processo de uma ou mais representações semânticas às árvores de derivação. Este processo pode ocorrer em uma fase posterior, ou à medida em que estas árvores são geradas.

Existem trabalhos sobre sistemas semânticos para interfaces de linguagem natural, capazes de interagir com diversos programas de aplicação.

[\[Voltar ao sumário\]](#)

2.2.4. Pragmática

À medida que se avança é necessário fazer uma interpretação do todo e não mais analisar o significado de suas partes, do ponto de vista léxico e gramatical.

As ligações anafóricas (exemplo: anáforas pronominais, como ele, este, etc.) e de outros fenômenos de referência merecem resolução.

Uma certa estrutura pode não acrescentar nenhum dado de interesse na interpretação de uma sentença, ao passo que um comentário sarcástico, como *Bela camisa Fernandinho*, pode ter significado distinto do que aparenta textualmente representar, inclusive até contrariando seu sentido literal. Crenças e contextos intencionais também são fatores que interferem em uma resolução semântica.

Outro problema pertinente ao processamento da língua natural é o tratamento da ambigüidade de estruturas complexas como anáforas, elipses, etc. (fenômenos pragmáticos textuais). Estes fenômenos são tratados através do cálculo dos significados implícitos, cálculo das relações interfrásicas, dentre outros.

[\[Voltar ao sumário\]](#)

2.3. Exemplo de processamento

Para elucidar os processos descritos anteriormente, será dado um exemplo extraído de [RIC 93], em que as etapas ocorrem sequencialmente. Cumpre salientar que os limites entre tais etapas são obscuros, havendo casos em que todas são executadas concomitantemente. O exemplo a seguir basear-se-á na seguinte frase:

Eu quero imprimir o arquivo .init do Mário.

- Análise Morfológica

Incumbe-se o analisador morfológico da tarefa de separar a expressão "do Mário" no substantivo próprio "Mário" e na preposição e artigo "de" e "o". Outrossim, este módulo deve reconhecer a sequência ".init" como uma extensão de um arquivo que está funcionado como adjetivo na frase.

Ainda nesta fase, todas as palavras são classificadas de acordo com suas respectivas categorias sintáticas.

- Análise Sintática

A análise sintática precisa explorar os resultados da análise morfológica para criar uma descrição estrutural da frase. O objetivo deste processo (*parsing*) é converter a lista de palavras que formam a frase em uma estrutura que defina as unidades representadas pela lista. No exemplo adotado, o resultado desta etapa está ilustrado na figura 2.1.

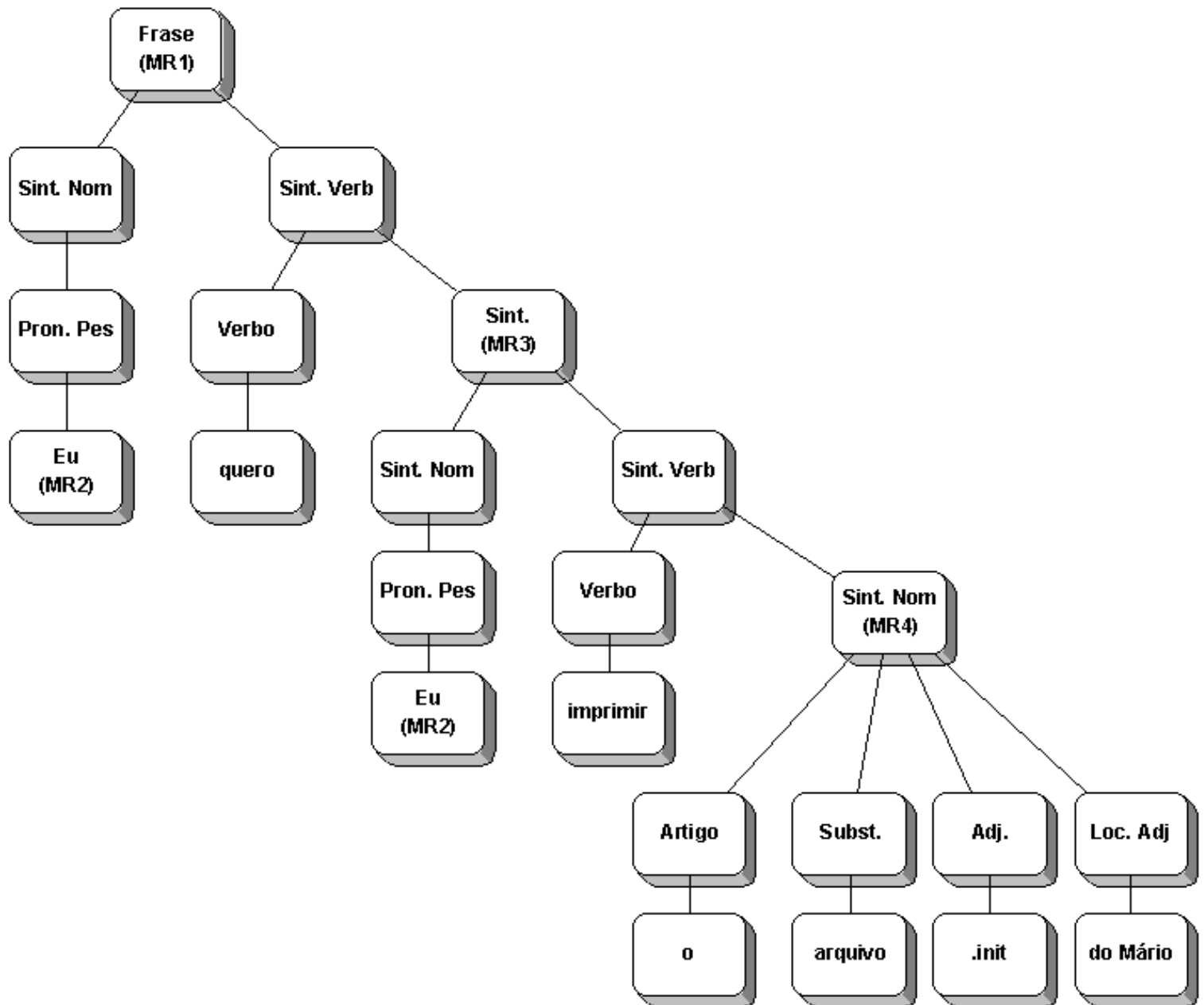


Figura 2.1 – Resultado da análise sintática.

Percebe-se que a frase foi convertida em uma estrutura hierárquica, contendo unidades significativas imprescindíveis à análise semântica. Também pode-se notar a presença de marcadores de referências – nem sempre utilizados em analisadores sintáticos – evidenciados na árvore sintática entre parênteses. Cada marcador refere-se a uma entidade mencionada na frase, configurando um componente ao qual será atribuído um significado na análise semântica.

• Análise Semântica

A análise semântica mapeia palavras isoladas para os objetos apropriados na base de dados e ainda cria estruturas corretas que correspondem ao modo como os significados das palavras isoladas combinam entre si.

Neste exemplo, supõe-se uma base de conhecimento baseada em *frames* que contém as unidades mostradas abaixo:

Usuário

É-um:	Pessoa
*nome-login:	Precisa ser <string>
Usuário068	
instância:	Usuário
nome-login:	Regina-Couto
Usuário073	
instância:	Usuário
nome-login:	Mário-Pires

F1

instância:	Estrut-Arq
nome:	coisas
extensão:	.init
dono:	Usuário073
diretório	/wspires

Estrut-Arq

É-um:	Informação-Objeto
-------	-------------------

Impressão

É-um:	Evento-Físico
*agente:	Precisa ser <animado ou programa>
*objeto:	Precisa ser <objeto-informação>

Desejo

É-um:	Evento-Mental
*agente:	Precisa ser <animado>
*objeto:	Precisa ser <estado ou evento>

Comando

É-um:	Evento-Mental
*agente:	Precisa ser <animado>
*executor:	Precisa ser <animado ou programa>
*objeto:	Precisa ser <evento>

Este-Sistema

Instância:	Programa
------------	----------

Com base nesta representação, pode-se gerar um significado parcial do exemplo, conforme explicitado abaixo:

MR1		{ toda frase }
instância:	Desejo	
agente:	MR2	{ Eu }
objeto:	MR3	{ evento de impressão }

MR2		{ Eu }
------------	--	--------

MR3		{evento de impressão}
instância:	Impressão	
agente:	MR2	{Eu}
objeto:	MR4	{arquivo .init do Mário}

MR4		{arquivo .init do Mário}
instância:	Estrut-Arq	
extensão	.init	
dono:	MR5	{Mário}

MR5		{Mário}
Instância:	Pessoa	
Nome:	Mário	

O marcador de referência MR1 corresponde ao evento de primeiro nível da frase. É um evento de desejo, onde a pessoa que fala, denotada pelo pronome "Eu", quer que ocorra um evento de impressão, no qual esta mesma pessoa que fala imprime um arquivo cuja extensão é ".init" e cujo dono é Mário.

• Integração de Discurso

Embora não tivesse sido explicado este processo, alguns autores [RIC 93] o classificam com relação ao fato do significado de uma frase isolada poder depender das frases que a antecedem e poder também influenciar os significados das frase que vêm depois dela.

No exemplo considerado, mesmo que já tivesse sido descoberto sobre o que se trata a frase, ainda não se sabe que indivíduos específicos estão sendo citados. Especialmente, não se sabe a quem se refere o pronome "Eu" ou o nome "Mário". Para esclarecer estas referências, recorre-se a um modelo de contexto do discurso, onde pode-se ficar sabendo que o usuário atual, o qual digitou a palavra "Eu", é o Usuário068 e que a pessoa chamada "Mário" é o Usuário073. Uma vez definido o referente correto de Mário, pode-se também determinar que arquivo está sendo mencionado: F1 é o único arquivo com extensão ".init" cujo dono é Mário.

• Análise Pragmática

Com o término das etapas anteriores, poder-se-ia registrar o que foi dito como fato e encerrar o processo. Em certas frases, cujo efeito pretendido é claramente declarativo, esta é precisamente a atitude correta. No entanto, em outras frases, como esta do exemplo, o efeito pretendido é diferente. Pode-se descobrir este efeito aplicando um conjunto de regras que caracterizam os diálogos cooperativos. Neste exemplo, considera-se que, quando o usuário afirma querer algo que o sistema é capaz de executar, então o sistema deve atender à solicitação. O significado final do exemplo está mostrado abaixo:

Significado:	
instância:	Comando
agente:	Usuário068
executor:	Este-Sistema
objeto:	P27

P27	
instância:	Impressão

agente: Este-Sistema
objeto: F1

A etapa final do processamento pragmático é traduzir, sempre que necessário, a representação baseada em conhecimentos para um comando, a ser executado pelo sistema. Neste caso, esta etapa é necessária, e vê-se que o resultado final do processo de compreensão é:

lpr /wspires/coisas.init

Onde "lpr" é o comando de impressão do sistema operacional considerado.

Todos os processos descritos são importantes em um sistema completo de compreensão da língua natural. No entanto, nem todos programas são escritos exatamente com esses componentes. Às vezes, dois ou mais são omitidos, resultando em um sistema mais fácil de criar para subconjuntos restritos de uma língua, mas mais difícil de ampliar no caso de uma abrangência mais extensa.

[\[Voltar ao sumário\]](#)

3. Implementação de um analisador sintático de sentenças da língua portuguesa

Antes de proceder-se à implementação de um processador de linguagem natural, algumas decisões precisam ser tomadas.

Neste capítulo, apresentar-se-ão as diretrizes que conduziram a proposta aqui encerrada, juntamente com a fundamentação correlata.

3.1. *Processamento simbólico da língua natural*

O *parser* proposto neste trabalho está baseado no processamento simbólico da linguagem natural, em contraposição às abordagens centradas em redes neurais. Inúmeras são as razões que elegeram o processamento simbólico a opção mais apropriada ao problema em questão. Em primeiro lugar, o processamento simbólico é inerentemente voltado à manipulação de regras rígidas, como são as regras gramaticais de qualquer língua natural. Como corolário deste fato, observa-se que as principais aplicações da programação simbólica costumam apresentar as seguintes propriedades:

- raciocínio baseado em objetivos;
- análise de causas;
- análise gramatical.

Além destas evidentes virtudes da programação simbólica, no que tange ao processamento de linguagem natural, as redes neurais não constituem-se em uma ferramenta adequada a tal tarefa, devido às suas limitações de processamento dependente de contexto. As redes neurais são muito eficazes no reconhecimento de objetos isolados, para cujo reconhecimento tenham sido treinadas de antemão. Em se tratando de línguas naturais, mais importante é a habilidade de aplicar regras dependentes de contexto, o que se percebe no processamento simbólico.

A despeito destas observações, que foram formalmente provadas por renomados pesquisadores, dentre os quais acha-se Marvin Minsky, há quem advogue o uso de redes neurais em processadores de linguagens naturais.

[\[Voltar ao sumário\]](#)

3.2. *Prolog como linguagem para processamento de línguas naturais*

PROLOG (PROgramming in LOGic) é a linguagem mais utilizada para a programação no paradigma lógico mas, afinal, por que implementar um processador de língua natural em PROLOG, preterindo linguagens como LISP, ou mesmo C?

PROLOG é uma linguagem ímpar, cujas propriedades vão ao encontro das necessidades que surgem quando da programação de sistemas processadores de línguas naturais. Arrolam-se abaixo as principais características que tornam PROLOG uma escolha incontestável:

- **Facilidade de modificação de estruturas de dados grandes e complexas:** Esta peculiaridade é de vital importância para o armazenamento de estruturas sintáticas, estruturas semânticas bem como de entradas léxicas, elementos presentes em programas que manipulam qualquer linguagem natural.
- **Capacidade de auto-análise e automodificação de programas:** O suporte à metaprogramação é muito forte em PROLOG, o que possibilita a adoção de modelos abstratos de programação.
- **Algoritmo *depth-first* embutido:** Para a busca das informações concernentes a um programa PROLOG (fatos e regras), o método de busca *depth-first* é internamente utilizado em sistemas PROLOG. Este algoritmo é de extrema utilidade para a implementação de analisadores sintáticos, sendo adotado pela maioria.
- **Incorporação de DCG (Definite Clause Grammar):** DCG é um formalismo que estende as gramáticas livres de contexto, possibilitando a identificação e utilização de inter-relacionamentos entre os componentes de uma regra gramatical. Além de cobrir a principal carência das gramáticas livres de contexto, as DCGs não oneram o processamento de sentenças de tamanho considerável, como o fazem as gramáticas sensíveis ao contexto. Deste modo, com as DCGs tem-se um formalismo de grande capacidade de expressão, aliada à eficiência. As gramáticas sensíveis ao contexto, quando empregadas no processamento de sentenças, denotam uma complexidade de tempo exponencial, em função do tamanho das entradas.

Em verdade, PROLOG foi originalmente projetada para o processamento de linguagem natural, eis por que as características supradiscriminadas lhe são inerentes.

As duas primeiras propriedades acima estão também presentes na linguagem LISP, ao passo que a linguagem C não evidencia nenhuma delas. Qualquer linguagem de programação pode ser adotada no trato de línguas naturais; porém, algumas facilitam mais do que outras.

[\[Voltar ao sumário\]](#)

3.3. Implementação

Como resultado deste trabalho, foi implementado um *parser* que gera a árvore sintática das sentenças válidas que lhe são submetidas. Obviamente, adotou-se um subconjunto bastante reduzido da língua portuguesa, porém, dadas as propriedades anteriormente descritas, o programa PROLOG resultante permite facilmente a adição de novas regras gramaticais e a agregação de novas palavras ao dicionário léxico.

Além de gerar a árvore sintática de sentenças corretas, é possível gerar todas as sentenças reconhecidas pela gramática definida, juntamente com as respectivas árvores sintáticas.

O código do analisador sintático denota todo o poder de processamento simbólico de PROLOG, linguagem que proporciona soluções compactas e elegantes para problemas deste tipo e para problemas recursivos em geral.

A seguir, apresentam-se alguns exemplos de utilização do analisador sintático:

```
?- sentenca(ArvSint,[o,comerciante,vende,batatas,velhas],[ ]).  
ArvSint=
```

```
sent(suj(art(o),subst(comerciante)),pred(vtd(vende),  
objdir(subst(batatas),adj(velhas))))
```

yes

```
?- sentenca(Arv,[o,comerciante,vende,batatas,velha],[ ]).  
no
```

```
?- sentenca(Arv,[eu,viajas],[ ]).  
no
```

```
?- sentenca(Arv,[tu,viajas],[ ]).  
Arv = sent(suj(ppret(tu)),pred(vti(viajas)))
```

yes

```
?- sentenca(ArvSint,[estes,comerciantes,vendem,morangos,velhos],[ ]).  
ArvSint=  
sent(suj(pdem(estes),subst(comerciantes)),pred(vtd(vendem),  
objdir(subst(morangos),adj(velhos))))
```

yes

```
?- sentenca(ArvSint,[estes,comerciantes,vendem,morangos,velhas],[ ]).  
no
```

```
?- sentenca(ArvSint,[estes,comerciantes,vendem,morangos,velho],[ ]).  
no
```

```
?- sentenca(ArvSint,[estas,comerciantes,vendem,morangos,velhos],[ ]).  
ArvSint=  
sent(suj(pdem(estas),subst(comerciantes)),pred(vtd(vendem),  
objdir(subst(morangos),adj(velhos))))
```

yes

[\[Voltar ao sumário\]](#)

4. Conclusões

A implementação de sistemas processadores de linguagens naturais, através de DCGs como formalismo gramatical, é deveras facilitada. As DCGs mostram um bom desempenho na análise de sentenças e, ademais, cobrem a principal carência das gramáticas livres de contexto, a qual é a incapacidade de estabelecer e utilizar inter-relações entre os componentes de uma regra gramatical. Como se não bastasse, analisadores sintáticos assim construídos podem ser facilmente estendidos através de truques semânticos de concepção simples. Destarte, as DCGs constituem-se em uma ferramenta que tende a tornar mais rápido o desenvolvimento de processadores de linguagens naturais.

[\[Voltar ao sumário\]](#)

Bibliografia Consultada

[ALL 95] ALLEN, James. **Natural Language Understanding**. The Benjamin/Cummings Publishing Company Inc, 1995.

[CLO 97] CLOCKSIN & MELLISH. **Programming in Prolog**. Springer-Verlag, 1997.

[COV 94] COVINGTON, M. **Natural Language Processing for Prolog Programmers**. Prentice-Hall, 1994.

[PER 94] PEREIRA, Fernando C., GROSZ, Barbara J. **Natural Language Processing**. MIT Press, 1994.

[RIC 93] RICH, E.; KNIGHT, K. **Inteligência Artificial**. Makron Books, 1993, 722p.

[STE 94] STERLING, L.; SHAPIRO, E. **The Art of Prolog**. MIT Press, 1994.

[VAN 93] VAN LE, T. **Techniques of Prolog Programming**. John Wiley & Sons, 1993.

[\[Voltar ao sumário\]](#)

FIM