UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES- CAMPUS FREDERICO WESTPHALEN
CURSO DE INFORMÁTICA

Acesso a Base de Dados Através da Linguagem Natural

Por

ELISA MARIA PIVETTA CANTARELLI

Trabalho de Conclusão de Curso
para obtenção do título de
Bacharel em Informática

Prof. Valmor José Prevedello
Orientador
Prof. Evandro Preuss
Co-orientador

Frederico Westphalen, julho de 1998.

Este trabalho é dedicado

Aos professores, colegas e funcionários da URI que tornaram agradável a nossa estada nesta Universidade

Ao professor e orientador, Valmor José Prevedello, que me auxiliou, não somente neste trabalho, mas em muitas outras atividades.

Em especial

Ao meu esposo Luiz e meus filhos Marlon e Margel, cujo amor, dedicação e compreensão sustentaram e estimularam toda esta caminhada

Sumário

Lista de Símbolos

IA	Inteligência Artificial
SN	Sintagma Nominal
SA	Sintagma Adverbial
SP	Sintagma preposicional
PLN	Processamento da Linguagem Natural
IBM	International Business Machine
NQL	Natural Language Query
KBNL	Knowledge Base Natural Language
4GL	Linguagem de Quarta Geração
G	Gramática
S	Sentença
SNC	Sintagma Nominal Complementar
SV	Sintagma Verbal
ATN	Augmented Transition Networks
LISP	Linguagem de programação baseada no processamento de listas
PROLOG	Linguagem de programação simbólica baseada no cálculo dos predicados

Lista de Figuras

Resumo

Explorar as possibilidades oferecidas pelos recursos computacionais ao processamento da linguagem humana tem sido um dos grandes desafios em nossos dias. A linguagem natural é primariamente processada por pessoas, de modo que, para projetar máquinas que a compreendam, devemos contar com o avanço das ciências cognitivas e com o desenvolvimento de técnicas adequadas. Existem várias tentativas de apresentar um modelo unificado, porém este ainda está além do alcance da pesquisa. Assim, nesse trabalho, procuramos mapear o mundo real da linguagem natural para o meio computacional, apresentando técnicas de reconhecimento conforme níveis lingüísticos e modelos de geração de gramáticas. Abordaremos também problemas e dificuldades com o processamento da linguagem que permanecem não resolvidos. Diante do exposto, nosso objetivo é, apesar da incógnita que é o processo de reconhecimento e processamento da língua, produzir uma interface através da linguagem natural que acesse bases de dados, utilizando ferramentas de quarta geração, mantendo uma relação de entendimento razoável com o usuário.

Abstract

To explore the possibilities offered by the computational resources to the processing of the human language has been one of the great challenges in our days. The natural language is processed primarily by people, so that, to project machines that understand it, we should count with the progress of the cognitive sciences and with the development of appropriate techniques. Several attempts exist of presenting an unified model, even so this is still besides the reach of the research. Thus, in that work, we tried to transpose the real world of the natural language for the computational way, presenting recognition techniques according to linguistic levels and models of generation of grammars. We will also approach problems and difficulties with the processing of the language that are not solved. Before the exposed, our objective is, in spite of the incognito that it is the recognition process and processing of the language, to produce an interface through the natural language that search bases of data, using tools of fourth generation, maintaining a relationship of reasonable understanding with the user.

Objetivos

Objetivos Gerais:

Desenvolver e implementar um sistema que possibilite ao usuário final interagir com uma base de dados (que suporta o padrão SQL) de forma que a entrada para consultas seja feita através da linguagem natural.

Objetivos Específicos:

- Realizar um estudo das necessidades disponíveis para tratamento da linguagem em nível de análise de sentenças;

- Implementar um software em linguagem de Quarta Geração, ou seja, Banco de Dados, para servir de suporte e viabilizar a interação com a linguagem natural ;

- Criar uma interface através da linguagem natural que possibilite ao usuário não especializado acessar a base de dados, elaborando suas próprias consultas;

1 Introdução

Muito antes da revolução industrial, o homem tem procurado aumentar os limites de suas habilidades, inventando aparatos mecânicos. A pá e a picareta, o carrinho de mão e outras ferramentas similares são exemplos que remontam a antigüidade. Mais recentemente, veículos a motor, robôs industriais e, finalmente, computadores têm sido acrescentados à elite dos assistentes artificiais. Mas, temos a linguagem que, com certeza, é o mais importante, o mais onipresente dos fenômenos sociais; é um pré-requisito para a própria existência das sociedades humanas.

O grande desafio então, é disponibilizar recursos que possibilitem o processamento desta linguagem através da computação. Neste trabalho sobre "**Acesso a base de dados através da linguagem natural**", pretendemos mostrar métodos que possibilitam a implementação de interfaces em Linguagem Natural.

Evidenciou-se recentemente, que várias são as técnicas que capacitam os computadores a ajudarem as pessoas a analisar problemas e a tomar decisões. Numerosas aplicações comerciais estão a caminho, despertando o entusiasmo das empresas, dos profissionais liberais e especialistas. Em resumo todo o ambiente, seja comercial, doméstico ou acadêmico se tornará mais racional. Em resposta, informações serão reunidas, sintetizadas e postas em forma útil o mais rápido possível.

Grande parte das pessoas, em especial os usuários ocasionais, poderiam fazer bom uso dos recursos disponíveis de software, solucionando muitos dos seus problemas, não o fazem por "medo" do computador, da complexidade e da diversidade das interfaces utilizadas. Os que se aventuram na tentativa de descobrir todos os segredos do sistema só o conseguem adaptando-se a interface oferecida pelo sistema.

Encontrar um método ideal para processar a linguagem natural é assunto que vem preocupando profissionais atuantes em vários campos do conhecimento, seja na ciência da computação, informática, tradução automática, robótica, engenharia do conhecimento e também áreas de - pesquisa específicas como a lingüística.

Para entender o processamento da linguagem natural através da computação, procedeu-se uma revisão bibliográfica, a literatura especializada, como Rich e Knight (1993), Harmon e King (1988), Date (1989), Keller (1991), Chomski (1971), Robin (1987), Kowaltowski (1983), Perini (1976) e Lobato (1986).

Para discutir aspectos relativos ao processamento da linguagem natural e sua utilização como interface a um banco de dados, formulam-se as seguintes questões que norteiam esta pesquisa:

1) Como identificar cada palavra de uma sentença, solicitada pelo usuário, e consequentemente verificar sua existência?

2) Como validar uma sentença sintaticamente?

3) Como traduzir uma sentença em linguagem natural para linguagem de máquina?

Este trabalho está organizado da seguinte forma: o segundo capítulo trás uma explanação sobre o que é linguagem natural e suas implicações, ambigüidade, gramaticalidade e aceitabilidade

No terceiro capítulo, apresentamos a área que trata do processamento da linguagem natural. O Enfoque é dado as áreas de aplicações e suas gerações.

Para a compreensão e processamento da linguagem natural é necessário entender os componentes do processo, os quais se encontram

no quarto capítulo, e dividem-se em: análise léxico-morfológica, análise sintática, análise semântica e pragmática. Ainda neste capítulo, explanamos modelos de gerações de gramáticas, seus formalismos e suas regras de produção.

No quinto capítulo, um breve enfoque a sistemas desenvolvidos com interface de linguagem natural, com alguns exemplos. No sexto capítulo, uma visão geral das linguagens, ambientes e instrumentos.

Para a concretização do trabalho, no sétimo capítulo relatamos o protótipo. A função deste protótipo é de receber uma sentença em linguagem natural, testá-la gramaticalmente e converter em comandos SQL (Structure Query Language) e consequentemente executá-la. Na implementação, o vocabulário será limitado a um universo específico, no caso, um cadastro de clientes.

Assim, explorar as possibilidades de realizar um tratamento sistemático da linguagem natural e disponibilizar meios para aproximar o conhecimento humano, nossa forma de linguagem com a computação, com objetivo de chegar a alguma forma de processamento automatizado, que apresente um modelo simples de acesso para os usuários não especializados, oferecendo uma interação homem-máquina que possibilite um tratamento natural é o objeto de estudo deste trabalho.

2 Compreensão da Linguagem natural

A linguagem destina-se à comunicação entre as pessoas, compreendê-la é extremamente complexo. O ser humano, desde tempos remotos, tem refletido sobre sua própria linguagem. Estas reflexões foram absorvidas por certas disciplinas: gramática, filosofia e lingüistica. Atualmente, a informática vem contribuindo com novas perspectivas capazes de revolucionar o estudo sobre a linguagem.

Ao dizermos que uma linguagem é natural, estamos considerando que é algo que já existe, e que provem de nossa interação com outras pessoas. Ao contrário de uma linguagem de programação formal ou artificial (Cobol, Pascal, C,...) que serve apenas a um propósito específico. A linguagem natural difere das linguagens formais, principalmente quando consideramos o conhecimento humano. O acervo de conhecimento da humanidade encontra-se expresso, basicamente, em linguagem natural, seja em meio magnético ou ainda em papel. [LIM 96]

O estudo da linguagem natural na informática, remonta ao próprio advento do computador. Despertou muitas esperanças, seguidas de muitas decepções. Decorreram muitos anos, até que os pesquisadores se convencessem de que o problema era a representação do conhecimento. Mesmo assim este problema ainda não foi resolvido, mas já é possível realizar com confiabilidade sistemas que têm desempenhos úteis de domínio de semântica ainda que restrito.

Quando tentamos compreender a linguagem falada, aproveitamos pistas, como gestos, expressões faciais, postura, às quais não temos acesso quando escrevemos um texto. Usamos a linguagem em uma variedade de situações tão ampla que nenhuma definição de compreensão é capaz de responder a todas elas. De forma que, para definir compreensão dentro da ótica de processamento através de máquina, devemos mapear a informação

recebida para uma forma mais diretamente utilizável. Quando iniciamos a tarefa de criar programas de computador que compreendem a linguagem natural, uma das primeiras coisas que temos que fazer é definir precisamente a tarefa subjacente e como deve ser a representação. Devemos assumir que nosso objetivo é ser capaz de raciocinar com o conhecimento contido nas expressões lingüísticas e explorarmos a forma ideal para processá-la.

Atualmente não existem dúvidas sobre a necessidade de se ter um modelo ideal para processar a linguagem humana através do computador. Entretanto, até hoje não existe uma área integrada, academicamente estabelecida, na qual se concentram os esforços referentes ao tratamento automatizado da linguagem natural. Também no Brasil, as pesquisas apresentam-se dispersas por diversas áreas, em geral sob a forma de dissertações, teses, comunicações e congressos ou simpósios, e publicações não-convencionais, de circulação restrita. [COU-KAY 91]

2.1 IA

Uma das áreas de grande relevância da Informática, é a Inteligência Artificial - IA (Artificial Intelligence - AI). Surgiu como disciplina científica a partir da Segunda Grande Guerra. Defini-la é um processo complexo, mas podemos dizer que é uma área que tenta simular a inteligência humana.

A Compreensão da Linguagem Natural é um dos campos de estudo da Inteligência Artificial. A IA possui outras áreas de aplicações, tais como:

- Sistemas Especialistas;
- Sistemas Inteligentes e autodidatas;
- Reconhecimento de Modelos;

- Educação Assistida por Computador;
- Sistemas Neurais.

- Sistemas Especialistas: para a programação organizada de árvores lógicas consecutivamente resolvendo preposições (silogismos), foram elaboradas linguagens próprias para a Inteligência Artificial, tais como LISP e PROLOG. O que não obriga que esse tipo de programas, conhecidos por Sistemas Especialistas (também conhecidos por Sistemas Expertos) sejam unicamente desenvolvidos nessas linguagens. No mercado atual encontramos programas deste gênero desenvolvidos nas mais diversas linguagens, tais como C, PASCAL e até mesmo BASIC. Abrangendo áreas como medicina, mecânica, etc., auxiliando os profissionais a executar mais rápido e com menor possibilidades de erros determinadas tarefas.

- Sistemas Inteligentes e Autodidatas: a partir da evolução do algoritmo utilizado em jogos como o conhecido "Jogo da Velha", surgiram os sistemas inteligentes e autodidatas. Ao contrário dos Sistemas Especialistas que necessitam de regras pré- estabelecidas, a decisão é própria. Discernindo as regras corretas das incorretas.

- Reconhecimento de Modelos :encontramos o reconhecimento de imagens e o de sons. É justamente a parte que mais tem evoluído e sido utilizada. A tendência é a exploração de todos os sentidos conhecidos, como a visão, audição, paladar, olfato e tato.

- Educação Assistida por Computador: a idéia de se ter num computador um instrutor inteligente é representada pela Educação Assistida por Computador, expondo questões correlacionadas ao desenvolvimento dos alunos.

- Sistemas Neurais: através de Sistemas Neurais, hoje muito comentados, é pretendido, baseado em sistemas neurológicos dos seres vivos, chegar a combinação de tudo que abrange a Inteligência Artificial.

Ainda hoje não se consegue criar uma inteligência tal qual a humana em máquinas. De qualquer forma, não podemos descartar a hipótese de que um dia isso possa acontecer. Encontramos diversos cientistas no mundo inteiro trabalhando e acreditando nesta possibilidade.

2.2 O que é Linguagem

"Linguagem é a utilização oral ou escrita da língua. Em tal sentido é que empregamos a palavra nas expressões Linguagem Oral e Linguagem Escrita. Num sentido mais genérico, linguagem seria qualquer sistema de sinais de que se valem os indivíduos comunicar-se."[AND 78]

Em meio a tanta diversidade de falas dentro de uma comunidade, é imprescindível manter-se a unidade da língua, pois graças a ela é que nós nos entendemos.

Uma língua pode ser definida como um conjunto de sentenças, sendo cada uma delas formada por uma cadeia de elementos (palavras ou morfemas). A estrutura das sentenças não se resume à colocação desses elementos em seqüência uns após aos outros, mas compreende também unidades intermediárias hierarquicamente dispostas.

Nem todas as combinações possíveis dos elementos de uma língua formam sentenças; mas, ainda assim, o número de sentenças possíveis em qualquer língua natural (isto é, português, japonês, inglês etc.) é infinito.

Pode-se conceber o conhecimento de uma língua como uma espécie de mecanismo que permite a formação e a interpretação de sentenças, e a lingüística como uma tentativa de deslindar as peças e o funcionamento desse mecanismo.

A par dos variados níveis de fala, existe uma linguagem padrão, utilizada por todos que buscam instrução e que, pelo aprimoramento cultural, necessitam expressar-se com mais clareza e precisão, de modo mais variado e fluente.

Porém, quando nos referimos a linguagem escrita, esta tem por finalidade representar a falada. Na medida do possível, pois os símbolos gráficos não conseguem evocar muitos elementos da fala. Apesar de suas limitações a linguagem escrita é de valor incalculável para a humanidade. Basta dizer que toda a cultura do homem tem sido preservada e perpetuada, através dos séculos, pela escrita. Neste particular, nenhum outro instrumento de comunicação se lhe pode comparar.

Dentro do enfoque da computação tanto a linguagem oral como a escrita, são objetos de estudo para desenvolver sistemas na área de Inteligência Artificial .

2.3 Implicações na Interpretação da Linguagem

Nenhum programa de linguagem natural pode ser completo porque novas palavras, expressões e significados podem ser gerados com bastante liberdade. A linguagem pode evoluir juntamente com a evolução das experiências que queremos comunicar. Ex.: Eu xeroco uma cópia para você.

Sabemos ainda, que existem inúmeras maneiras de dizer a mesma coisa. Como na frase abaixo:

- Maria nasceu no dia 11 de outubro

ou

- O aniversário de Maria é no dia 11 de outubro.

As frases de uma língua são descrições incompletas das informações que pretendem transmitir. A linguagem permite aos que falam ser tão vagos ou precisos quanto quiserem. Ela também permite que se deixe de mencionar coisas que achamos que os ouvintes já sabem. Ainda, a linguagem nos permite dizer coisas sobre um mundo infinito usando um número finito de símbolos.

Um dos grandes debates filosóficos deste século priorizou à questão do que significa uma frase. A compreensão da linguagem envolve seu mapeamento para alguma representação que seja apropriada a uma determinada situação .Fica então difícil responder o que significa uma frase, e o que é compreensão da linguagem. Usamos a linguagem em uma variedade de situações tão ampla que nenhuma única definição de compreensão é capaz de responder por todas elas.[RIC 94]

2.3.1 Ambigüidade

Um processo complicado no caso da compreensão da linguagem são as frases genuinamente ambíguas, como: O pé está úmido.

Se faz necessário encontrar não apenas uma interpretação, mas todas possíveis, ao contrário não poderemos definir que pé está úmido. Poderia ser, o pé da laranjeira, o pé da mesa ou o meu/seu pé. Ou ainda, “a manga é vermelha”.

Porém, mais complicado ainda é quando não existe nenhum caso de homonímia lexical que possa constituir a base da sua ambigüidade. Na frase, “a cortina de seda antiga” , encontramos os dois seguintes sentidos:

- a cortina de seda em si que é antiga (evidente, neste caso, que a seda também é antiga)

- a seda é que é antiga (podendo a cortina em si ter sido fabricada recentemente, embora não necessariamente)

As duas estruturas de constituintes imediatos serão pois:

a)

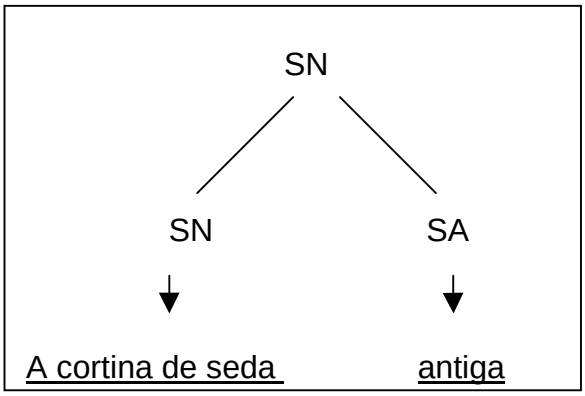

Figura 2.1 - Sentença gerando ambigüidade
Fonte - Perini – Gramática Gerativa

b)

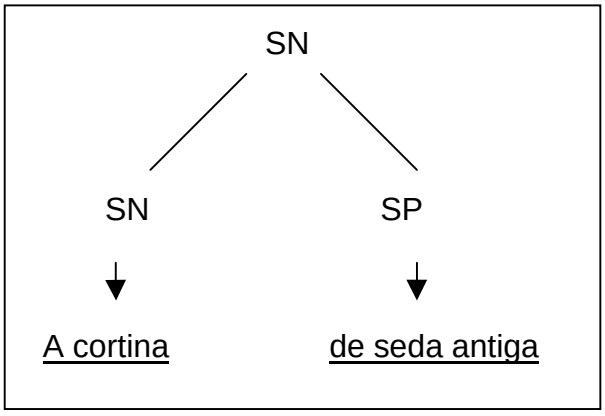

Figura 2.2 – Sentença gerando ambigüidade
Fonte - Perini – Gramática Gerativa

Onde:
SN → sintagma nominal
SA → sintagma adverbial
SP → sintagma preposicional

2.3.2 Gramaticalidade e aceitabilidade

É importante diferenciar gramaticalidade de aceitabilidade, que estão de certa maneira relacionadas e notar que não são determinadas unicamente pelo nosso conhecimento da língua. Este é apenas um dos fatores relevantes, ao lado de limitações de memória, atenção e falta de atenção, estado de espírito e elementos do contexto situacional, como barulho, número de interlocutores e assim por diante. Todos esses fatores podem interferir no funcionamento do nosso mecanismo lingüístico, isto é, no uso que fazemos do nosso conhecimento da língua.

Existem seqüências de elementos que são aceitas pelos falantes da língua com enunciados normais, ao passo que outras não são. Exemplo:

o livro está aberto
o está aberto livro

A primeira seqüência é aceitável, enquanto a Segunda seqüência é inaceitável.

Aceitabilidade é um fenômeno essencialmente intuitivo: muitas vezes, ao encontrarmos uma sentença como a segunda, “sentimos” que ela não é normal em português. Essa intuição não é resultado do nosso estudo de português na escola, porque qualquer falante da língua, ainda que nunca haja freqüentado uma escola, pode emitir julgamentos deste tipo.

O que nem sempre é fácil é decidir se determinada seqüência de elementos de uma língua é aceitável ou não. Os julgamentos de todos os falantes acerca das duas sentenças anteriores são claros, mas há casos em

que o falante hesita em seu julgamento, ou em que diferentes falantes da mesma língua têm julgamento diversos.
Exemplo:

> 1 - “Os conspiradores planejam incendiarem o Parlamento.”
> 2 - “Os conspiradores planejam constantemente incendiarem o Parlamento.”
> 3 - “Os conspiradores planejam diariamente em seu esconderijo no depósito de bondes da capital incendiarem o Parlamento.”

Segundo Perini a frase inaceitável seria a 1, e que 2 é melhor que 1 e 3 ainda melhor que 2, sem que no entanto nenhuma dessas seqüências seja tão aceitável quanto a 4:

> 4 – “Os conspiradores planejam incendiar o Parlamento.”

Ainda, é possível que alguém discorde a respeito de 1, 2 e 3, o que seria mais um exemplo ilustrando esta explicação. [PER 76]

É evidente que não podemos considerar gramaticais apenas as sentenças que já foram efetivamente observadas. A língua é um conjunto infinito de sentenças, e portanto mesmo que tivéssemos uma lista completa de todos os enunciados até hoje proferidos não poderíamos elaborar a partir daí uma lista completa de sentenças gramaticais dessa língua. Conforme já apontado anteriormente, estamos a todo momento produzindo ou recebendo frases totalmente novas.

2.4 Contribuições

Várias áreas têm contribuído para soluções no estudo da linguagem. A neurologia, a filosofia, a metafísica, a psicologia e a lingüística são algumas que merecem destaques. Mas quando as pesquisas deslocam-

se para a representação do conhecimento os psicólogos têm mais a contribuir do que os lingüistas.

Quando os especialistas em informática querem construir um sistema no qual se faz uso a linguagem natural, a primeira reação é voltar-se para os lingüistas, ou para as vagas lembranças de gramática, datando sua própria vida escolar. É considerável, pois os resultado obtidos pelos lingüistas constituem um conhecimento extremamente valioso. Porém, uma língua delimita em primeiro lugar a um conjunto que corresponde aquilo que é aceitável e a outro o não aceitável. De outro lado, deve-se considerar que o conhecimento jamais é o suficiente, e nem sempre necessário, para os objetivos fixados pela informática.

Do ponto de vista da psicologia, o que realmente envolve é toda a análise referente ao sentido, significado, levando em conta o comportamento humano. Este, muitas vezes, converte a linguagem natural em uma linguagem “interior”. Numerosas pesquisas, mostram que um texto é sempre processado, antes de ser memorizado, ao ponto de as pessoas se revelarem incapazes de designar, entre várias versões semanticamente equivalentes, aquela que efetivamente leram.

Na informática, não é obrigado respeitar regras, sejam elas na área da lingüística ou da psicologia, ainda que, seja interessante segui-las. Não é necessário justificar os métodos nem no plano formal nem no da conformidade com os resultados experimentais. “O único gabarito segundo o qual o especialista poderá vir a ser julgado é o da eficácia (rapidez, portabilidade e capacidade de extensão) dos algoritmos utilizados para resolver os problemas. Naturalmente, modelos e hipóteses antecipados por psicólogos e lingüistas são úteis e bem vindos.”[HAR 88]

3 Processamento da Linguagem Natural

O Processamento da Linguagem Natural (PLN), é uma área da Inteligência Artificial(IA) que têm por objetivo dotar as interfaces de computadores da capacidade de comunicar-se com seu usuário na língua deste.

Para processarmos a linguagem natural, duas etapas são necessárias, a primeira é a compreensão e a segunda é a produção. A compreensão, envolve reconhecimento com exatidão da linguagem, exigindo etapas de análise. Um sistema para processar a linguagem natural normalmente reúne módulos associados aos níveis lingüísticos de processamento de linguagem, os quais serão descritos no próximo capítulo.

As técnicas desenvolvidas para o processamento de linguagens artificias devem ser ampliadas e expandidas a vários campos de estudo. A linguagem natural é primariamente processada por pessoas e para conseguirmos projetar máquinas que a compreendam plenamente, devemos contar com o avanço da lingüística e das ciências cognitivas, como a psicolingüística.

Na década de 50, no Massachussets Institute of technology, começavam a ser implementados trabalhos que definiam as linguagens artificiais. Em 1956 a IBM implementava o FORTRAN. Na mesma época pesquisas avançavam em relação à lingüística nas ciências naturais, através de trabalhos desenvolvidos por Noam Chomsky. Chomski redefiniu a lingüística num modelo mais formal. Estudou regras de estrutura da frase baseando-se numa estrutura profunda, o léxico e regras transformacionais, buscando explicitar o funcionamento mente-cérebro. Obtendo assim, “uma gramática universal”, a qual estaria por traz de qualquer língua.

Atualmente, vários elementos têm contribuído para impulsionar os estudos nesta área. Pesquisas no tratamento computacional da

linguagem natural vêm sendo desenvolvida principalmente em três aplicações:

- **Interfaces Homem-máquina:** Interfaces em linguagem natural, com o objetivo de atender principalmente aos usuários não especializados, por apresentarem simplicidade da linguagem necessária têm alcançado resultados práticos na implementação.

- **Tradução Automática:** Implementações que possibilitam a tradução automática de textos entre diferentes idiomas. A dificuldade de criar sistemas de qualidade se deve, a falta de formalismos adequados para o tratamento do significado do texto reconhecido.

- Recuperação de Informações: Possibilita usuários a escrever, consultar banco de dados, utilizando a linguagem natural, sendo do sistema a responsabilidade da tradução para uma linguagem computacional.

3.1 Tecnologia do Processamento: Interface Homem-máquina

A tecnologia do processamento da linguagem natural em relação a interface homem-máquina, divide-se em duas gerações:

- Sistema Query de Base de Dados em Linguagem Natural – NQL (Natural Language Query);
- Sistemas de Linguagem Natural Baseados no Conhecimento – KBNL (Knowledge Base Natural Language).

3.1.1 Query em Linguagem Natural

Query em Linguagem Natural (NQL), é a primeira geração de sistemas de linguagem natural cujo conhecimento específico está

incorporado em suas regras para compreensão de queries. Sua função principal é servir como front-end em resposta a uma query em linguagem natural.

O acesso por meio da linguagem natural é um importante avanço. Usuários com pouco ou nenhum conhecimento podem interagir com a máquina, permitindo acesso a vários arquivos sem que o usuário tenha que aprender uma linguagem query formal, como as de Quarta geração (4GL).

Um usuário poderia querer selecionar no cadastro de clientes, como por exemplo:

“todos os clientes que possuem renda maior que R$1.000,00 e ser do sexo Masculino”,

Num ambiente 4GL a consulta seria:

```
Select nome, salário, fone
From cliente
Where salário > 1000
And
Sexo = Masculino
```

Figura 3.1 – Exemplo de uma consulta em SQL

Para um usuário que não tem conhecimento de linguagens de quarta geração é improvável que ele consiga realizar este tipo de consulta, mas se houver a possibilidade da entrada ser em forma de linguagem natural, mesmo seguindo algumas regras, esta consulta poderia ter sucesso.

Apesar das vantagens, não é necessário usar um sistema NQL por muito tempo para perceber que ele não “entende” uma pergunta da mesma forma que um assistente humano entenderia. São muito limitados no

poder de entendimento, em parte porque sua meta principal é de traduzir um query numa forma aceitável e os dados disponíveis são os fatos armazenados em sua base de dados.[KEL 91]

Muitas vezes operar com sistemas NQL pode ser irritante. Embora sejam programados há evitar respostas erradas, a compreensão é limitada e geralmente impede de fornecer algo mais que uma resposta correta e razoavelmente adequada. É irritante pois usuários finais esperam que qualquer sistema que se diz capaz de entender a linguagem natural, entenda não somente as palavras mas também a intenção por trás delas. Infelizmente os sistemas NQL não têm habilidade de compreender a real intenção de um usuário ao formular uma pergunta.

3.1.2 Linguagem Natural Baseada no Conhecimento

Um sistema baseado no conhecimento pode responder todas as questões de um mesmo tipo, como um sistema NQL, mas ele vai um passo adiante. Ele possui conhecimento extensivo de um domínio, incluindo as razões das pessoas fazerem certas perguntas e como ser útil em áreas de incerteza.

Em muitas situações query, não há informação suficiente na pergunta ou na base de dados para permitir uma compreensão profunda da questão, a despeito dos esforços para ser capaz de fornecer ao usuário uma resposta expressiva.

A tecnologia KBNL faz isso mantendo trilhas, numa base de dados, das muitas situações que podem possivelmente ocorrer de forma sensata num domínio específico. Estes dados não estão na base de dados , são situações possíveis para um dado domínio chamado Scripts. É por meio deste scripts que o sistema é capaz de compreender a intenção do usuário

quando este formula sua questão ou de agir segundo as implicações de uma declaração.

A real compreensão significa ter um conhecimento completo de um contexto particular. Quase toda a situação que encontramos possui scripts mais ou menos bem definidos. Na vida real, somos capazes de manter bons relacionamentos porque entendemos os scripts da situação na qual nos encontramos. O conhecimento especializado em uma dada situação geralmente depende de quão bem se entende as suposições não declaradas naquela situação.[KEL 91]

O conhecimento KBNL é bastante similar a um sistema especialista, embora as metas do sistema especialista sejam um tanto quanto estreitas. O objetivo principal de um sistema especialista é executar tarefas de planejamento e tomada de decisão de forma análoga a que um especialista humano faz; quando sua base de conhecimento chega a um estágio de verdadeiro especialista, espera-se ter a opção de substituir o especialista humano. Os sistemas KBNL estão envolvidos também com o planejamento e tomada de decisões baseadas no conhecimento. Entretanto, eles explicitamente adicionam a isto a habilidade de comunicar em linguagem natural compreensiva e são mais prováveis de serem usados como um aprendiz inteligente ou consultor.

4 Gramáticas e Analisadores

Para a compreensão e processamento da linguagem natural é necessário entender os componentes do processo, os quais se dividem em:

- **Análise Léxico-Morfológica**: realiza um tratamento a nível das palavras, permitindo reconhecer as palavras sob as diferentes formas que sua função na sentença lhes confere. O analisador léxico-morfológico identifica, numa sentença as palavras ou expressões elementares da língua, e obtém, para cada uma delas, as diferentes categorias em que podem estar atuando, com outras informações disponíveis através do léxico.

- **Análise Sintática**: O analisador sintático funciona com um parser. Utilizando gramática da linguagem a ser analisada, e uma seqüência de informações provenientes da análise léxico-morfológica, a respeito das palavras, busca construir árvores de derivação para cada sentença (onde são explicitadas as relações entre as palavras que compõem a sentença), determinando a gramaticalidade ou não da sentenças. Seqüências lineares de palavras são transformadas em estruturas que mostram como as palavras são relacionadas entre si. Algumas seqüências podem ser rejeitadas, se violarem as regras da linguagem. Ex. Maria a vai cinema ao

- **Análise Semântica**: As estruturas criadas pelo analisador sintático recebem significado. É feito um mapeamento entre as estruturas sintáticas. As seqüências lingüisticas cujo sentido o analisador semântico deve calcular compõem-se de um certo

número de palavras, identificadas pela análise morfológica, e reagrupadas em estruturas de análise sintática.

- **Análise Pragmática**: A estrutura que representa o que foi dito é reinterpretada para determinar o que realmente quis dizer. A análise semântica se restringe, normalmente, a lidar com os significados das sentenças tais quais são determinados pelos significados de suas partes, integrando o ponto de vista léxico e o gramatical. Mas normalmente a compreensão não ocorre assim, por partes. À medida que avançamos, vamos construindo uma interpretação do todo. Isso exige a resolução das ligações anáforas e de outros fenômenos de referência. Os fenômenos pragmáticos-textuais (como a co-referência, anáfora associativa, elipse, etc.) devem ser tratados através do cálculo das relações interfrásicas, do cálculo das referências e do cálculo dos significados implícitos, tentando levar em conta, inclusive, os atos da fala.

Todos os componentes vistos são importantes em um sistema que processe a linguagem natural, mas nem todos os programas são escritos exatamente com esses componentes, às vezes dois ou mais são omitidos. Esta omissão resulta em um sistema mais fácil de ser criado, porém mais difícil de ampliar no caso de uma abrangência mais extensa. [RIC 94]

O processo de compreensão de uma frase poderá ser um grande processo de busca, muitas vezes exaustivo. Há necessidade de decidir se todos os caminhos serão explorados ou se apenas um, e com este produzir o resultado. Em todos os processos de busca emerge o problema de decisão, qual caminho ou quantos devem ser seguidos e como trabalhar.

A primeira etapa de qualquer sistema é procurar palavras em um dicionário (o léxico, que compõe o processamento semântico) e extrair seu significado. Infelizmente, para fazer o mapeamento de palavras em uma

base de conhecimento, a ambigüidade léxica é maior do que denota no “dia-a-dia”. Como no exemplo já visto: “o pé está úmido ”. Se todas as formas de interpretação for considerada, pode se tornar muito dispendioso e caro. Muitos sistema tratam de uma única interpretação plausível e se rejeitada faz-se uma nova tentativa, resultando um sistema prático e satisfatório.

4.1 Processamento Morfológico

A utilização de processamentos morfológicos evita que se tenha de mencionar no léxico todas as formas possíveis que uma palavra pode assumir. Conserva-se uma forma única, como num dicionário, devendo encontrar-se as demais por meio de regras que descrevem as flexões possíveis. O léxico torna-se mais organizado, ficando mais rápido e menos volumoso.

O analisador morfológico tem a função de achar, a partir da forma constante do texto, a forma representativa de uma palavra armazenada no léxico. A grande dificuldade é que o léxico detém as informações quanto a natureza gramatical dificultando o processamento.

Uma maneira de contornar a situação é considerar todas as desinências a *priori,* da mais curta a mais longa*,* após examinar se para cada uma dessas desinências existe uma raiz no léxico, verificando ainda se há concordância de natureza gramatical entre a palavra encontrada e a respectiva desinência. Todas as soluções satisfatórias são conservadas e a ambigüidade encontrada será tratada em níveis superiores.

Outro problema do analisador morfológico é que os módulos morfológicos não podem ser construídos de forma desejável. Alguns proponentes de sistemas julgam necessário uma análise particular para cada caso, acrescenta-se a isto o caráter efêmero das regras e a falta de unanimidade sobre o que é o uso correto.

Há também usos que não apresentam regularidade suficiente. O verbo cozer, por exemplo, cozimento gera uma ação e cozido gera o resultado da ação. Contudo, não existe uma regra na língua portuguêsa que nos permita classificar em “ação” os substantivos que terminam em “mento”.

4.2 Processamento Léxico

O léxico ou dicionário, importante estrutura de dados que acompanha o processo de análise e geração da linguagem natural, armazena as palavras e associa, às mesmas, informações. Pode conter todas as palavras ou ser estruturado contendo apenas morfemas. A vantagem do primeiro, é podermos reconhecer e processar diretamente a palavra sem necessidade de reconstruí-la. O segundo, por sua vez , permite uma estrutura mais organizada.

“O nível léxico é responsável pela leitura do texto, caracter a caracter, identificando as palavras, onde uma palavra é um conjunto de caracteres que determinam uma unidade de informação. Faz então, corresponder a palavra, uma vez reconhecida às informações de que se dispõe sobre a mesma.” [SUE 97]

Seja qual for a aplicação, o emprego do léxico é obrigatório. O léxico fornece as mais diversas informações associadas as palavras, que permitem analisar e compreender a sentença em seu todo. Essas informações referem-se principalmente:

- ao fato de que cada palavra deve ser considerada no nível de sua aplicação;
- à codificação da palavra sob uma outra forma (como, um código escolhido para delimitá-la melhor);
- às informações gramaticais: natureza da palavra, suas flexões etc.;

- às informações semântico-pragmáticas, tais como número, lista de características, indicadores de sinônimos, etc.,

Todos esses itens dependem da linha de trabalho adotada. Em informática lida-se com grandes quantidades de informações. Por isso, a extensão dos léxicos não acarreta maiores dificuldades. Em alguns casos léxicos pequenos já são suficientes. É ilusão querer catalogar, de uma só vez, o conjunto de todas as palavras úteis de uma aplicação. Convém prever uma organização que possibilite atualizar os léxicos, logo que os mecanismos de diálogo ou de compensação permitam recuperar as informações complementares a serem associadas às novas palavras assim inseridas.

Permite ainda, a associação das informações disponíveis à palavra reconhecida, como a codificação para a palavra, informações gramaticais e informações semânticas. Para facilitar o processo na análise léxica, é empregado o analisador morfológico que antecede o processo de análise léxica.

O processo de determinar o significado correto de uma palavra isolada é chamado de “desambiguação” do sentido da palavra ou “desambiguação” léxica. Associam-se a cada palavra do léxico informações sobre os contextos que pode aparecer cada sentido da palavra. Cada uma das palavras de uma frase pode fazer parte do contexto em que os significados das outras palavras precisam ser determinados.

4.2.1 Processamento léxico-morfológico

O analisador léxico-morfológico identifica, numa sentença, as palavras e obtém para cada uma, as diferentes categorias em que podem estar atuando, com outras informações disponíveis do léxico. Por exemplo, a palavra *um* poderá ser analisada como um artigo indefinido ou como um numeral. A riqueza da análise dependerá da riqueza do léxico utilizado.

A ambigüidade léxica-morfológica ocorre quando uma mesma palavra denota entidades diferentes do mundo, normalmente em famílias diferentes. A nível léxico é importante que todas as formas possíveis sejam buscadas e devolvidas pois a ambigüidade somente poderá ser tratada em níveis mais avançados.

4.3 Processamento Sintático

O processamento sintático é a etapa em que a frase plana é convertida em estrutura hierárquica que corresponde às unidades de significado da frase. Embora haja sistemas que pulem esta etapa, ela é muito importante. Se não houver análise sintática, o sistema semântico terá que decidir quais são seus próprios componentes. Quando a análise é feita, por outro lado, restringe o número de componentes que a semântica pode examinar. A análise sintática é menos dispendiosa em termos computacionais do que o processamento semântico. Assim, ela pode desempenhar um papel significativo na redução da complexidade do sistema.

A análise sintática trata da disposição das palavras, reconhecidas pelo analisador léxico, numa sentença. É responsável pela criação de uma árvore de derivação, onde são explicitadas as relações entre as palavras que compõe o texto.

Durante a construção da árvore de derivação é realizada a verificação da adequação da seqüência de palavras, os termos que compõe a frase, período ou oração, às regras de formação de concordância, regência e posicionamento dos termos impostos pela linguagem.

Para tanto, a análise sintática utiliza uma representação gramatical, onde estão declarados os fatos sintáticos da linguagem, as

estruturas de composição dos sintagmas e um procedimento de análise, responsável por verificar se a frase de entrada esta de acordo com a regras gramaticais e gerar uma estrutura hierárquica, representando a estrutura da frase analisada.

O desejo de se construir sistema modulares, faz com que se encontre freqüentemente em informática uma divisão de tarefas semelhante aquela dos lingüistas. A arquitetura na maioria dos sistemas existentes, se limita a uma execução seqüencial dos processamento, do morfológico ao pragmático, com possibilidades, ainda que limitadas, de voltar atrás. Ao mesmo tempo que se reconhecem as vantagens de uma abordagem mais integrada, as realizações práticas permanecem raras.

“Os modelos que utilizam uma sintaxe se caracterizam pelo fato de reconhecerem que as sentenças têm uma estrutura. Estes modelos se adaptam à descrição da linguagem natural, mas convém lembrar que em geral, não se prestam muito à análise das línguas.” [COU 92]

Existem muitas maneiras de produzir uma análise, quase todos os sistemas possuem pelo menos uma representação declarativa, chamada de gramática, dos fatos sintáticos da linguagem. Um procedimento, chamado analisador, que compara a gramática com as frase de entrada para produzir estruturas analisadas.

Conforme Chomsky, grande pensador na área da lingüística, existe por trás da língua, de um modo não palpável, um corpo de generalizações, princípios e regras abstratas em número finito, que determinam as frase da língua, a sua gramaticalidade, suas propriedades e características. Este corpo altamente organizado chama-se gramática. Cada ser humano possui então uma gramática interiorizada, adquirida enquanto criança num período relativamente curto e possivelmente na base de alguns princípios inatos, próprios à espécie humana, “a faculdade da linguagem”.

Formalmente, uma linguagem é um conjunto de sentenças, onde cada sentença é a concatenação de um ou mais símbolos, chamados palavras, do vocabulário da linguagem. Os lingüistas formalizaram um conceito de gramática em seus estudos sobre a linguagem natural e definiram como sendo a especificação finita deste conjunto.

Os formalismos das gramáticas fundamentam muitas teorias lingüísticas, que por sua vez dão base para muitos sistemas de compreensão da linguagem natural. Entretanto, nas teorias da lingüística moderna, Chomsky (1986), Gazdar (1985), Sells (1986) e Bresnan (1982) concordavam que o processamento da linguagem natural têm pouco em comum com os sistemas de processamento de linguagem para computador (como os compiladores) do que era de se esperar. As gramáticas sendo elas puras ou livres de contextos não são eficazes ou adequadas para descrever as linguagens naturais.

Entretanto, o esforço se concentrava em definir uma sentença válida e então, prover descrições estruturais com o objetivo de tornar possível o “entendimento” de uma linguagem pelo computador.

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

Em uma gramática simples o nó da árvore gramatical corresponde a uma palavra da frase de entrada ou a um símbolo não-terminal na nossa gramática. Cada nível corresponde à aplicação de uma regra da gramática.

A gramática de uma língua é uma descrição finita, capaz de gerar todas as sentenças desta língua e apenas estas. Uma linguagem formal construída com base em um alfabeto, também chamado de vocabulário

terminal, é um subconjunto de todas as cadeias formadas pelos elementos deste alfabeto. A operação de base é chamada reescrita, que permite substituir um axioma **S** por cadeias compostas de elementos terminais e/ou auxiliares, sendo estes últimos representados por letras maiúsculas.

As gramáticas são objetos de diferentes classificações em função de sua complexidade. Caso a linguagem seja composta por um conjunto finito de sentenças a gramática pode ser uma simples listagem de todas as sentenças válidas. Por outro lado, para as linguagens finitas, é necessário um formalismo mais complexo.

No primeiro caso, o das linguagens finitas, o analisador pode ser implementado como um simples algoritmo de pesquisa sobre o conjunto das sentenças válidas. No segundo caso, o das linguagens infinitas, que correspondem a maioria das linguagens utilizadas, o analisador é um algoritmo que caminha sobre a representação gramatical da linguagem determinando se a sentença pertence, ou não, a linguagem.

Alguns conceitos básicos são necessários à uma introdução nesta área, são eles:

Gramática Gerativa: número limitado de regras a partir do qual se pode gerar um número infinito de frases que formam uma língua, dando-lhe uma caráter aberto, dinâmico e criativo.

Frase: é uma unidade de linguagem que comunica um pensamento ou a intenção de uma pessoa.

Sintaxe: é o estudo das regras que determinam quais cadeias de palavras de um vocabulário podem formar frases.

“Nem toda cadeia de palavras sintaticamente correta é uma frase.”[LOB 86]

Para ilustrar o conceito de gramática gerativa, vamos supor um vocabulário limitado a um par de letras {a,b}. E frases bem formadas que respeitam a sintaxe da linguagem, seriam as seguintes:

Figura 4.1 – Exemplo de uma gramática gerativa

onde, a Segunda metade é igual à primeira invertida. As regras sintáticas que geraram estas frases podem ser as seguintes:

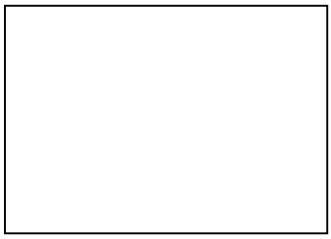

Figura 4.2 – Regras sintáticas que podem gerar a figura 4.1

Estas regras geram as frases conforme o seguinte procedimento: inicia-se com o símbolo *S* e o substituí pelo lado direito da regra, se este lado também possuir um *S*, repete-se o processo até que não haja mais *S*. Por exemplo:

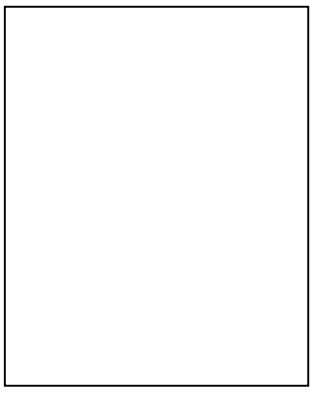

Figura 4.3 – Gramática Gerada a partir das regras da figura 4.2

Os elementos escritos em minúsculo são chamados *terminais* (no caso *a* e *b*), ou seja, partindo somente deles não é mais possível aplicar-se qualquer regra. Enquanto os escritos em maiúsculo são *não-terminais* (no caso *S*).

A partir deste exemplo, pode-se ilustrar como formalmente uma gramática (G). Ela é composta por quatro conjuntos de elementos. Assim:

G = (N,∑,P,So), onde
N - conjunto dos não terminais
∑ - conjunto dos terminais
P - regras gramaticais ou de produção
So - é um não-terminal que serve como símbolo inicial

Por exemplo:

G = ({ S } , { a,b } , { S→aa, S→aSa, S→bb, S→bSb } , So)

A linguagem gerada por esta gramática é representada como:
L (G) = (conjunto de todas as frases de G)

As regras de produção podem ser formadas de muitas maneiras, caracterizando seu poder de expressão e consequentemente implicando na complexidade da linguagem. Chomsky classificou as gramáticas como sendo de quatro tipos: linear, livre de contexto, sensível ao contexto e irrestrita. A complexidade das regras, e o esforço computacional necessário para manuseá-las é significativamente diferente para cada tipo. Também o ferramental matemático necessário para prover o tratamento formal de cada tipo é significativamente mais elaborado de um tipo para outro. Estas classes estão contidas propriamente umas nas outras. Pode ser observado que uma gramática irrestrita engloba uma gramática sensível ao contexto que, por sua vez, engloba uma gramática livre de contexto que engloba a linear.

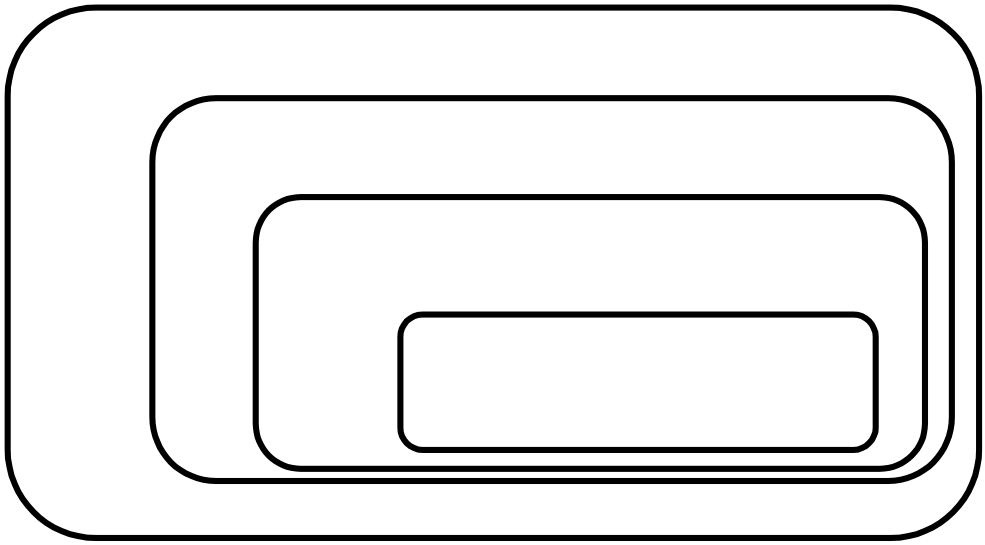

Figura 4.4 – Classificação das Gramáticas segundo Chomski

As gramáticas que "produzem" essas linguagens são chamadas de gerativas. Cada categoria é capaz de gerar a classe correspondente, e as nela contidas. As gramáticas regulares e livres de contexto são insuficientes, por exemplo, para a partir delas produzir frases como : "Ana, Maria e Paula são as esposas de José, Mário e Mauro, respectivamente". Por outro lado, as gramáticas sensíveis ao contexto se tornam ineficientes, por introduzirem ambigüidade e por conterem regras complexas e de difícil leitura.

4.3.1 Gramática Linear

Gramáticas regulares são bastante simples e facilmente reconhecidas, porém tem um poder de expressão limitado, consequentemente não muito úteis para o processamento sintático da linguagem natural.

As regras de derivação são do tipo **Y→αX** , onde **α** designa um elemento do alfabeto, e **X** e **Y** são símbolos que pertencem ao vocabulário auxiliar. **X** pode estar ausente. As linguagens descritas por essas gramáticas caracterizam-se pela possibilidade de acrescentar, um número arbitrário de

vezes, uma seqüência de palavras, desde que previsto pela respectiva gramática.

$X \rightarrow \alpha Y$ ou $X \rightarrow \alpha$

X e Y são não terminais e α é um terminal

Gramática:

S →aX

S → bY

Y → a

Y → aX

X → b

X → bY

Exemplo:

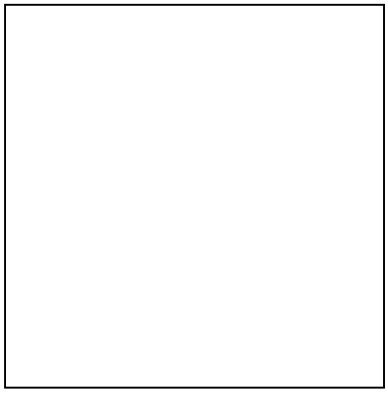

Figura 4.5 - Gramática linear

Onde:

GN = grupo nominal

N = grupo nominal sem determinante

SGN = seqüência do grupo nominal

d = determinante

n = nome

prep = preposição

cc = conjunção coordenativa

Sentença e árvore de derivação:

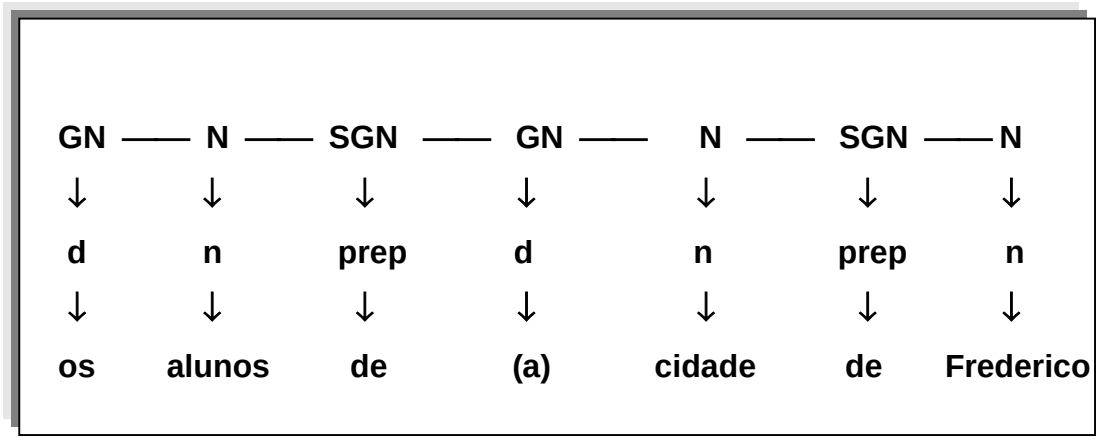

Figura 4.6 – Árvore de derivação -gramática Linear
Fonte-COULON-KAYSER – Informática e Linguagem natural

Esta mesma gramática pode ser representada, como segue:

GN → d N
N → n | n SGN
SGN → prep GN | cc GN | prep N

O símbolo | serve como abreviatura, a saber: A → α | β resume em uma única representação as duas regras A → α e A → β.

4.3.2 Gramática Livre de Contexto

Todas as regras de produção de P tem o seguinte formato

X → α

Onde X está em N e α está em $(N \cup \Sigma)^+$
$(N \cup \Sigma)$ é a união dos Não terminais com os terminais;
+ o conjunto resultante tem um ou mais elementos
* o conjunto resultante tem zero ou mais elementos

Permitem especialmente resolver os chamados fenômenos de encaixe. Por essa operação, inclui-se totalmente uma seqüência em outra, inserindo-a no lugar de um dos constituintes dessa última.
Gramática:

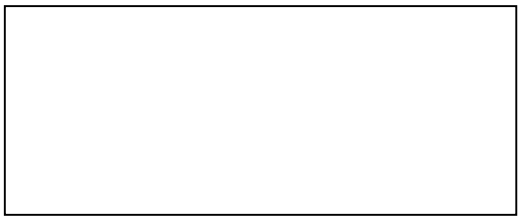

Figura 4.7 – Gramática Livre de Contexto

Gera seqüências como:

S → bX
→ babXb
→ babab

A questão se as gramáticas de contexto livre bastam para descrever as línguas naturais é assunto controvertido. Após Ter sido contestada tal possibilidade, a questão foi reaberta a partir dos trabalhos de Salkoff -79 e de Gazdar – 83 . Entretanto, o número de regras consideradas por esses autores é tão grande (da ordem de bilhão), que as conseqüências práticas são quase as mesmas.

Deve-se assinalar, contudo, que modelos de contexto livre e com um pequeno número de regras são perfeitamente adaptados a subconjuntos muito limitados da linguagem natural.

4.3.3 Gramática Sensível ao Contexto

Se todas as regras de produção de P possuem o seguinte formato:

$Y_1 \; X \; Y_2 \rightarrow Y_1 \; \alpha \; Y_2$

Onde, X pertence a N; Y_1, Y_2 e α estão em $(N \cup \Sigma)^+$. Intuitivamente, X pode ser restrito com α, mas somente no contexto de Y_1 e Y_2.

Gramática:

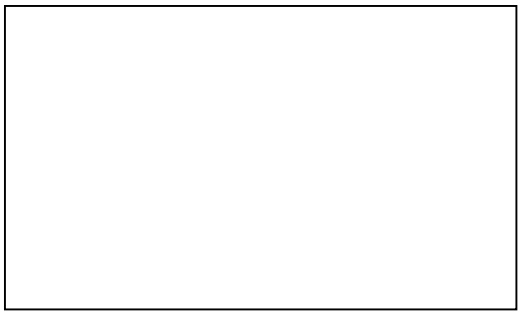

Figura 4.8 – Gramática Sensível ao Contexto

Onde a seguinte seqüência pode ser gerada:

S	→	aX_1bc
	→	abX_1c
	→	abX_2bcc
	→	aX_2bbcc
	→	aaX_1bbcc
	→	$aabX_1bcc$
	→	$aabbX_1cc$
	→	$aabbX_2bccc$
	→	$aaX_2bbbccc$
	→	aaabbbccc

Em uma gramática deste tipo as regras somente podem ser utilizadas em contextos onde o lado esquerdo, incluindo seus terminais e não-terminais, “casa” com o conjunto que se deseja provar. Na gramática definida para o exemplo anterior o não-terminal X1 só pode ser substituído se for seguido por c (regra b) ou por b (regra e).

4.3.4 Gramática Irrestrita

Se todas as regras de produção de P apresentam a seguinte forma: $\alpha \rightarrow \beta$

onde, α pertence ao conjunto $(N \cup \sum)^+$ e β pertence a $(N \cup \sum)^*$.

Figura 4.9 – Gramática irrestrita

Grande parte das implementações de verificadores de sintaxe, como os utilizados em compiladores, queries de banco de dados, e até mesmo para a linguagem natural, se utilizam gramáticas do tipo livre de contexto. Isto se deve ao fato desta gramática ter uma performance muito melhor devido à sua maior simplicidade de implementações e utilização e porque atende aos principais requisitos exigidos por estas aplicações.

4.4 Parsers ou Árvores de Derivação

Na geração de sentenças para gramáticas lineares e livres de contexto podemos observar que a partir do símbolo inicial So, outros símbolos são “chamados” até que a seqüência de caracteres da entrada esteja completa. Este processo pode muito bem ser representado por uma árvore, onde So é a raiz e os não terminais (X,Y,....) da regra So seus ramos. Os ramos (X,Y) por sua vez, podem ter outros ramos, até chegarmos as folhas da árvore, os símbolos terminais.
Dado a gramática

S → aa | aSa | bb| bSb, com a seqüência abbaabba, temos a seguinte árvore.

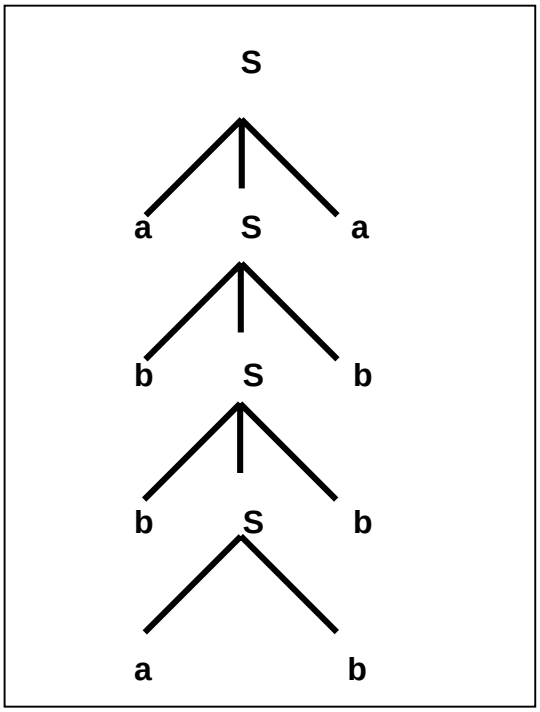

Figura 4.10 – Parsers ou árvore de derivação

Consideramos uma gramática, onde:

N = { S, SN, SV, SP }

S = Sentença

SN = Sintagma Nominal;

SNC= Sintagma Nominal Complementar;

SV = Sintagma Verbal e

SP = Sintagma Preposicional.

∑= { artigo, substantivo comum, substantivo próprio, numeral, verbo, prep.}

P = { S → SV SNC

SV → Verbo SN

SN → Artigo SubstComum SN

SN → Artigo SubstComum SP

SP → Preposição SN

SP → preposição

SNC→SubstPróprio

}
So = S

Podemos interpretar P da seguinte forma: um SV é formado por um verbo seguido de um SN. Um SN é formado por artigo seguido de um substantivo, seguido de um SP ou do próprio SN. A mesma interpretação pode ser feita para as demais regras.

Deve ser observado que $\sum$ é formado por categorias de palavras, por exemplo:

Artigo = {o,a,os,as}

Substantivo comum = { cliente, endereço,cidade, telefone}

Substantivo próprio = { Frederico Wesphalen, Seberi,Maria}

Verbo = { selecionar, mostrar, listar}

Preposição = {para, de, por}

Conjunção = { com, e}

Derivada desta gramática, poderia ser geradas frases como:

- Mostre os clientes de Frederico Westphalen.
- Liste o nome, o telefone dos clientes de Seberi

O exemplo a seguir mostra a derivação da árvore para as frases citadas acima.

a) Mostre os clientes de Frederico Westphalen.

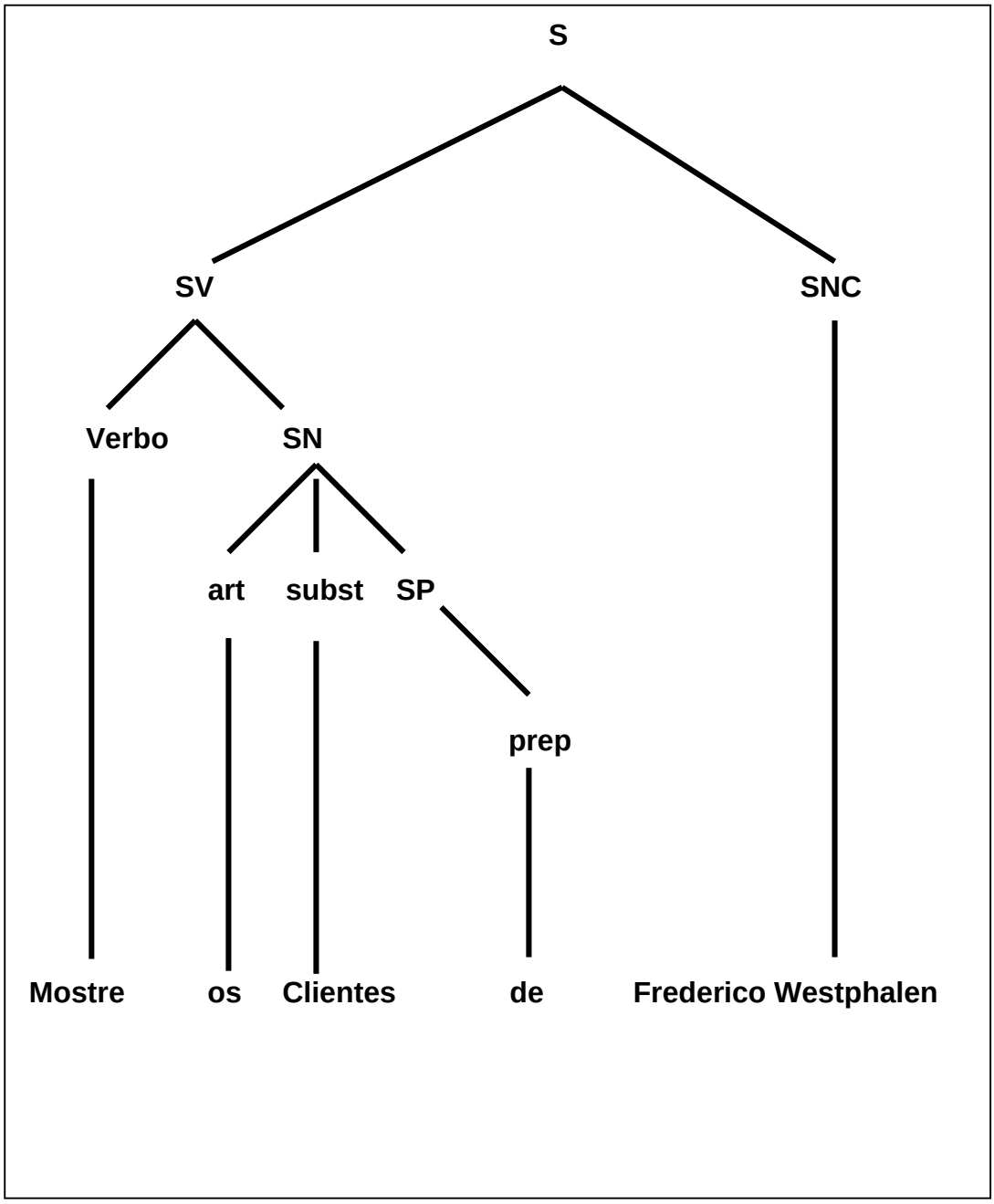

Figura 4.11 – Exemplo da árvore de derivação - sentença 1

b) Liste o nome, o telefone, dos clientes de Seberi

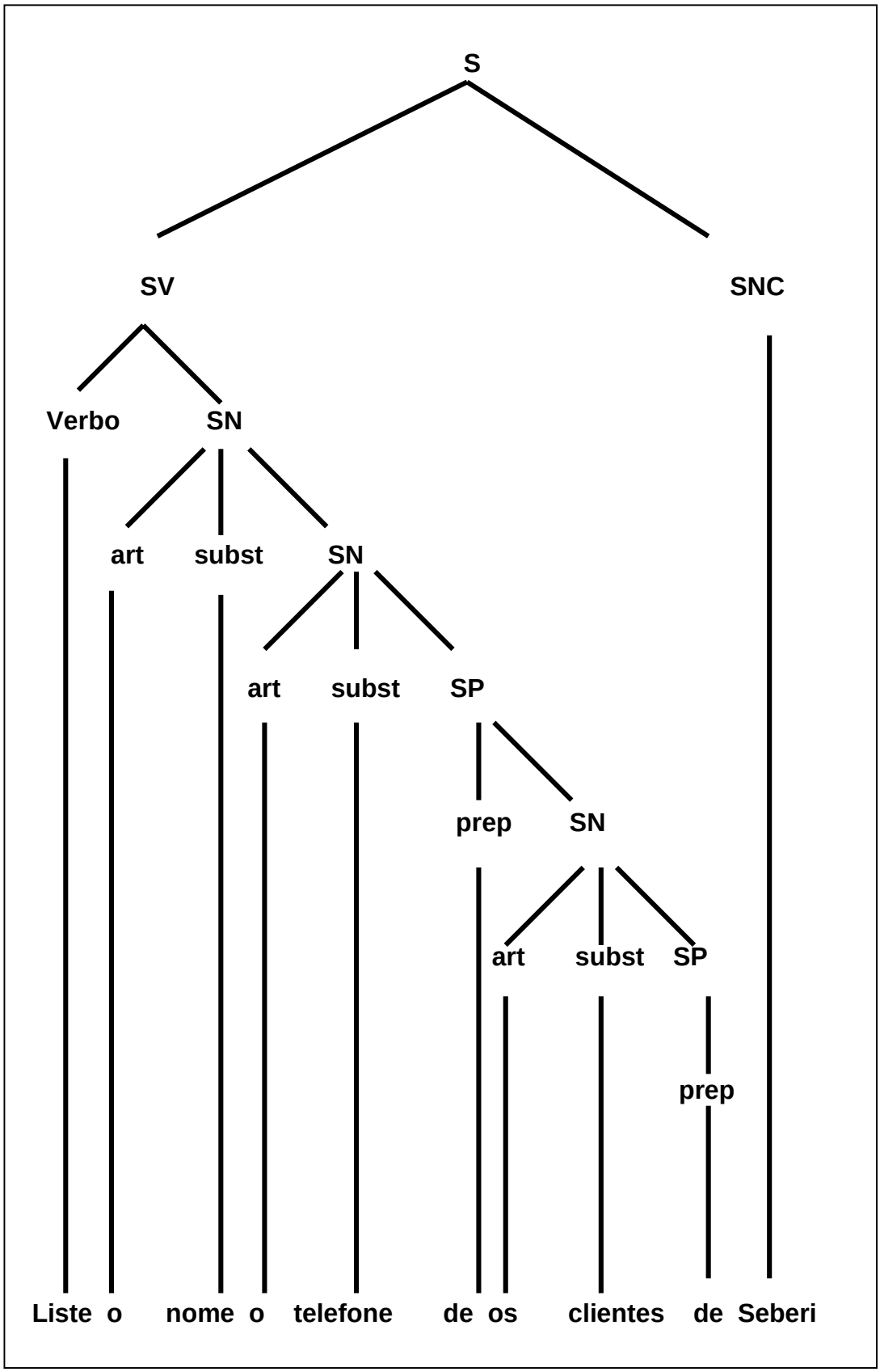

Figura 4.12 – Exemplo da árvore de derivação - sentença 2

Dado um G=(N, ∑, P, So), o procedimento para criar uma árvore de derivação a partir das regras pode ser facilmente descrito como:

Se $S_n \rightarrow X_1\ X_2\ X_3\ ....X_n$ pertencentes a P, então liga-se a raiz Sn aos nódos $X_1\ X_2\ X_3\ ...X_n$ como sendo seus descendentes.

- Para cada descendente, $X_i \rightarrow Y_1\ Y_2\ Y_3\ ...Yn$, liga-se X_i aos nódos $Y_1\ Y_2\ Y_3\ Y_n$ como sendo seus descendentes.

- Continue até que todos os conjuntos de descendentes sejam terminais, ou strings vazias.

4.5 Modelos que não requerem uma sintaxe

Um modelo que não utiliza a sintaxe, apesar de ser muito restrito, mas com mérito da simplicidade, possui duas categorias de palavras: as portadoras de significado e as demais. Esta abordagem leva a reduzir o texto a um conjunto de palavras chaves. Os sinônimos processados permitem identificar se um texto responde positivo ou não a uma consulta formulada. Métodos assim, não podem ser empregados num domínio semântico muito extenso, pois se existirem muitas palavras o risco de ambigüidade é grande, ou quando as construções forem mais complexas. Existem aplicações satisfatórias que utilizam este método.Com efeito, não seria nada descabido traduzir uma consulta desta forma: ***Mostre o endereço de Maria*** por ***endereço, Maria***. No entanto, convém acrescentar fatores limitativos, como a ordem das palavras (para distinguir, por exemplo***, a casa do chefe*** de ***o chefe da casa***).[COU 92]

4.6 Processamento Semântico

Há sempre dois componentes de uma linguagem qualquer: a maneira na qual ela é escrita – sua sintaxe e o significado que contém – sua semântica.

No processamento da linguagem natural o problema semântico é crítico e deve ser tratado em todos os níveis do processo de reconhecimento da linguagem. Pode-se dizer que todas as etapas do tratamento envolvem um forte componente semântico.

A nível léxico, a categorização das palavras, em categorias gramaticais, provê um primeiro nível de tratamento semântico. Ao associarmos informações como gênero, número, a uma palavra, estamos atribuindo à mesma um significado, que será fundamental para o tratamento semântico da linguagem.

A nível sintático, temos o conceito de dependência, por exemplo dependência verbal onde o sujeito deve concordar com o verbo. Esta dependência, ao ser tratada como um problema gramatical, é na verdade um conjunto de restrições semânticas impostas pela linguagem para determinar se uma sentença é, ou não, bem formada.

A análise sintática-semântica da linguagem natural assemelha-se muito aos conceitos aplicados às linguagens de programação, onde a sintaxe é uma expressão da forma das sentenças válidas, enquanto a semântica se preocupa com uma teoria dos tipos, rejeitando as construções válidas, porém sem sentido.

Então, o primeiro mito a desfazer é o de que existe um conjunto rigoroso de regras que formam a gramática de qualquer linguagem, o que não é verdade, pois uma linguagem, tal como o inglês ou qualquer outra, é suficientemente sutil para fazer com que nenhum conjunto simples de regras

seja suficiente para descrever todas as maneiras em que se pode dizer alguma coisa. Outra idéia importante é que nem tudo o que obedece mesmo as mais simples regras da gramática faz sentido. Por exemplo, sonho dorme furiosamente, é uma sentença corretamente formada em português, tendo um substantivo, um verbo e até um advérbio, mas não tem nenhum sentido. “Sonho” é um substantivo, mas não é algo que possa “dormir”, que por sua vez, é um verbo ao qual não é razoável aplicar o advérbio “furiosamente”. Em resumo, uma sentença correta mas sem sentido. [SUE 98]

Várias abordagens para o problema da criação de uma representação semântica para uma frase foram desenvolvidas, incluindo as seguintes:

- Gramáticas de casos, onde a estrutura criada pelo analisador contém informações semânticas.

- Gramáticas semânticas, que combinam conhecimento sintáticos, semânticos e pragmáticos em um único conjunto de regras na forma de uma gramática.

- Análise conceitual, os conhecimentos sintáticos e semânticos são combinados em um único sistema de interpretação orientado pelo conhecimento semântico. Nessa abordagem, a análise sintática está subordinada à interpretação semântica.

- Interpretação semântica aproximadamente composicional, onde o processo semântico é aplicado ao resultado de uma análise sintática.

4.6.1 Gramática de Caso

As gramáticas de caso, introduzidas por Fillmore 68, resolvem um certo número de problemas, por abordarem diferente o problema de como

combinar a interpretação sintática e semântica. As regras gramaticais são escritas para descrever regularidades sintáticas, e não semânticas. Mas as estruturas que as regras produzem correspondem as relações semânticas, e não estritamente sintáticas.

Consideramos as sentenças: *Maria lava a louça* e *Carla lava a louça* têm a mesma estrutura. Como diferem apenas pelo sujeito, parece lícito fundi-las em uma só, ou seja, *Maria e Carla lavam a louça***.** Essa transformação já não é possível com a sentença *O sabão lava a louça*, pois a composição *Maria e o sabão lavam a louça* não é aceitável.

Uma transformação desse tipo deve ter como condição que os dois sujeitos sejam da mesma classe, o que não se verifica com Maria e sabão. [COU 92].

Outro exemplo, *mamãe cozinhou por três horas* e *o feijão cozinhou por três horas***.** A estrutura sintática das duas frases são quase idênticas. O que muda é somente o sujeito, "mamãe" e "o feijão". É importante notar que essa informação semântica na verdade interfere na sintaxe da língua. Embora seja permitido juntar duas frases paralelas, ou seja, podemos unir, como por exemplo, "*o feijão e o arroz cozinharam*", mas nunca " *A mamãe e o feijão cozinharam*".

Não existe uma clara concordância em torno de qual deve ser exatamente o conjunto correto de casos profundos, mas alguns óbvios serão listados a seguir:

(A) Agente - Instigador da ação
(I) Instrumento - Causa do evento ou objeto usado para causar o evento
(D) Dativo - Entidade afetada pela ação
(F) Factitivo - Objeto ou ser resultante do evento
(L) Locativo - Local do evento
(F) Fonte - Local a partir de onde alguma coisa se move
(M) Meta - local para onde alguma coisa se move

(B) Beneficiário - Ser em cujo interesse (ou benefício) o evento ocorreu

(T) Tempo - Tempo em que ocorreu o evento

(O) Objeto - Entidade que recebe a ação ou que se transforma, o caso mais genérico. [RIC 94]

O processo de análise em representação de casos é dirigido pelas inserções léxicas associadas a cada verbo. As linguagens têm regras para o mapeamento de estruturas de casos subjacentes para formas sintáticas de superfície. Por exemplo, se A está presente , ele é o sujeito. Caso contrário, se I estiver presente, ele é o sujeito. Ou então, o sujeito é O.

A análise través da gramática de casos normalmente é dirigida por expectativas. Uma vez localizado o verbo da frase, ele pode ser usado para prever os sintagmas nominais que ocorrerão e determinar o relacionamento desses sintagmas como resto de frase.

As redes de transição aumentadas (ATNs), que serão descritas nas próximas páginas, oferecem uma boa estrutura para análise através da gramática de caso. Ao contrário dos algoritmos de análise tradicionais, em que a estrutura resultante sempre espelha a estrutura das regras gramaticais que a criaram, ATNs permitem estruturas de saída de formato arbitrário.

4.6.2 Gramática Semântica

"Uma gramática semântica é uma gramática livre de contextos, onde a escolha de símbolos não-terminais e regras de produção é governada por funções semânticas e sintáticas. O resultado da análise e da aplicação de todas as ações semânticas associadas ao significado da frase. Essa ligação íntima entre ações semânticas e regras gramaticais funciona porque as próprias regras gramaticais são criadas em torno de conceitos semânticos chave."[SUE 97]

Para evitar as dificuldades em uma análise sintática completa, antes de iniciar qualquer interpretação, ou aquelas dificuldades que surgem

ao se associarem os dois níveis (sintáticos e semânticos) ao longo de uma análise, deve-se ampliar o mecanismo de análise das gramáticas de contexto livre. Classifica-se as palavras não apenas segundo sua natureza gramatical, mas também pelo seu significado.

Suas principais vantagens são:

- Quando a análise estiver completa, o resultado pode ser usado imediatamente, sem o estágio adicional do processamento do que seria necessário caso uma interpretação semântica ainda não tivesse sido realizada.

- Ambigüidades que surgem durante uma análise estritamente sintática podem ser evitadas. Considere como exemplo: “Eu quero imprimir todos os alunos na impressora X”. Durante uma análise estritamente sintática, não seria possível decidir se o adjunto adverbial “ na impressora X” refere-se a “quero” ou a “imprimir”. Se usarmos uma gramática semântica, não haverá ambigüidade na interpretação.

- A frase: “Mostre todos os carros que sonham” pela análise sintática a sentença estaria correta mas quando passar pela semântica resultaria em erro. A semântica acusa que a ação “sonhar” esta associada a um ser humano, portanto carro não é humano e não sonha.

- As questões sintáticas que não afetam a semântica podem ser ignoradas. Ex.: Imprimir todos os alunos de Frederico Westphalen na impressora X.

No entanto, há certas desvantagens no uso das gramáticas semânticas:

- Números de regras exigidas pode ficar muito grande, porque muitas generalizações sintática não são feitas.

- Devido ao grande número de regras de gramática, o processo de análise pode ser dispendioso.

- Depois de muitas experiências com as gramáticas semânticas, em uma variedade de domínio, a conclusão parece ser de que elas são úteis para produzir rapidamente interfaces de subconjuntos restritos de linguagem natural. Mas, não se acomodam como solução global para o problema de compreensão da linguagem. Elas não foram aprovadas por não conseguirem captar importantes generalizações lingüísticas.[RIC 94]

4.7 Processamento Pragmático e do Discurso

Para compreendermos uma frase, mesmo que simples, é necessário considerar o contexto do discurso e o contexto pragmático em que a frase foi preferida. Estas questões tornam-se mais importantes, quando temos por finalidade a compreensão de diálogos.

Considere o texto:

Meu carro foi assaltado ontem.
Eles levaram o aparelho de som e os CDs.

O pronome “eles” deve ser reconhecido como referência aos ladrões que entraram na casa.

Para ser possível reconhecer o relacionamento entre as frases, o conhecimento sobre o que está sendo discutido tem que ser muito grande. Os programas que executam a compreensão de várias frases recorrem a grandes base de conhecimento ou a forte restrições sobre o Domínio do discurso que apenas uma base de conhecimentos mais limitada seja

necessário. A maneira como esse conhecimento é organizado é crucial para o sucesso do programa de compreensão.

Os processamentos efetuados nos diferentes níveis são bastantes específicos. As realizações práticas de processar através de máquina todos os níveis e chegar a um bom resultado, ainda que restrito, são raras. Os dois primeiros níveis não oferecem maiores dificuldades. Este porém, não é o caso dos níveis superiores, semântico e o pragmático.

4.8 Redes de transição aumentadas (ATNs)

Desde o primeiro uso da ATN (Augmented Transition Networks) no sistema LUNAR (Woods, 1993), que permitiu acesso a um grande banco de dados de informações sobre a geologia lunar, o mecanismo tem sido explorado em muitos sistemas de compreensão da linguagem.

As ATNs são formas de representação para gramáticas de linguagens naturais, em substituição às gramáticas formais. As ATNs aceitam alguns modelos gramaticais existentes (sintáticos-semânticos). São grafos do tipo redes recursivas de transição, de que são um incremento. Estão baseadas nos mesmos autômatos finitos que emergem do processamento das gramáticas regulares, mas com acréscimo de testes adicionais e efeitos colaterais, na forma de ações apostas a seus arcos. Com tal expansão, esses mecanismos tornam-se suficientemente poderosos para processar as linguagens naturais até o grau de manuseio exigido pelas tarefas que se esperam atualmente dos computadores.

Os acréscimos feitos às redes recursivas para torná-las ATN incluem:

- registros de informações sobre árvores de derivação parciais entre saltos para diferentes sub-redes;

- os arcos da rede, além de conter nomes de classes gramaticais ou estruturas sintáticas, podem ter associados a eles teste arbitrários que devem estar satisfeitos antes que eles possam ser seguidos numa transição;

- ações especiais podem ser apostas a um arco, para que sejam executadas toda vez que aquele arco seja seguido.

As ATNs são, ao contrário das gramáticas irrestritas, meios práticos de implementação dos métodos de geração e reconhecimento de linguagens naturais. Outra vantagem das ATNs é a possibilidade de incluir em sua representação diversas características que teoricamente pertencem a diferentes formas gramaticais. Elas podem ser usadas, entre outras coisas, para testar novas idéias sobre as teorias gramaticais, existente ou sob pesquisa.

Os testes e as ações opostos aos seus arcos tornam as ATNs bem providas para manipular os encargos das gramáticas transformacionais: casos especiais e exceções, tão bem como as regras gerais estão em perfeito controle. A principal desvantagem da ATN quando aplicadas ao processamento da linguagem natural, é sua dificuldade em manipular enunciados agramaticais, embora significativos. Isso decorre de sua forte dependência das estruturas sintáticas.

A implementação desses princípios dependem fortemente do sistema computacional, sua capacidade de dados e sua velocidade, e também da linguagem utilizada.

O processo de análise das ATNs é descrito como sendo a transição de um estado inicial para um estado final em uma rede de transição . Foi estendida uma classe de marcações que podem ser agregadas aos arcos que definem as transições entre os estados. Os arcos podem ser marcados com uma combinação de:

- Palavras específicas, como "com".

- Categorias de palavras, como "verbo".

- Extensões para outras redes que reconhecem os componentes significativos de uma frase. Por exemplo, uma rede destinada a reconhecer um sintagma preposicional (SP) pode incluir um arco se estenda para um sintagma nominal (SN).

- Procedimentos que criam estruturas que farão parte da análise final.

Para melhor entendimento segue um exemplo, elaborado por Kevin-Elaine (1994), de como uma ATN funciona na frase "*O grande arquivo foi impresso*".

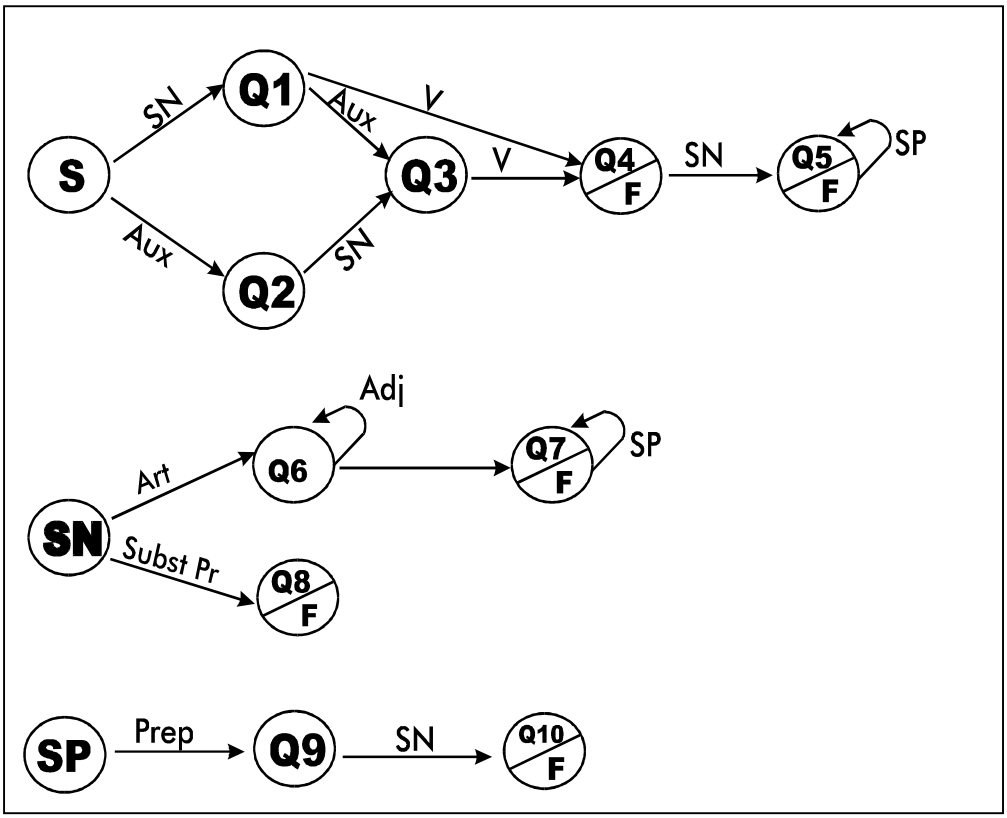

Figura 4.13 - Exemplo em ATN
Fonte: Rich e Knight - IA

As etapas dessa execução são:

1. Comece com o estado F.
2. Vá para SN.
3. Teste para ver se "o" é um artigo.
4. Se o resultado do teste for positivo, atribua ao registro *artigo,* o valor *definido* e vá para o estado Q6,
5. Teste a categoria para ver se "grande" é um adjetivo.
6. Se positivo, anexe "grande" à lista contida no registro Loc Adj. Fique no estado Q6.
7. Teste a categoria para ver se "arquivo" é um verbo. Esse teste falha.
8. Teste para ver se "arquivo" é um substantivo. Se sucesso, atribua ao registro substantivo o valor "arquivo" e vá para o estado Q7.
9. Estenda para SP.
10. Teste a categoria para ver se "foi" é uma preposição. Falha, então sinalize fracasso.
11. Não há nada mais a ser feito no estado Q7. Retorne. O retorno faz com que máquina vá para o estado Ql,
12. Teste para ver se “foi" é verbo. Se sucesso, então atribua NIL ao registro AUX e atribua ao registro V “foi". Vá para o estado Q4.
13. Vá para o estado SN. Como a próxima palavra "impresso" não é um artigo ou substantivo próprio, FN retornará fracasso.
14. A única outra coisa a fazer no estado Q4 é parar. Mas ainda restam outras informações, então não foi encontrada uma análise completa. O retrocesso se faz necessário.
15. O último ponto de escolha estava no estado Ql, então volte para lá. Os registros AUX e V precisam ser redefinidos.
16. Teste para ver se “foi" é um verbo auxiliar. Sucesso, então atribua ao registro AUX para o valor "foi" e vá para o estado Q3.
17. Teste para ver se "impresso" é um verbo. Sucesso, portanto atribua ao registro V o valor “impresso”. Vá para o estado Q4.
18. Como todas as entradas se esgotaram, Q4 é um testado final aceitável. Retome a estrutura.

Existem várias maneiras de usar ATNS que não foram mostradas neste exemplo:

O conteúdo dos registros pode ser trocado. Por exemplo, se a rede fosse expandida para reconhecer orações passivas, no ponto de detecção do passivo, o conteúdo atual do registro SUJ seria transferido para um registro OBJ e o objeto da preposição "por" seria colocado no registro SUJ. Assim, a interpretação final das duas orações a seguir seria a mesma:

Maria deletou o arquivo
O arquivo foi deletado por Maria

"Testes arbitrários podem ser colocados nos arcos. Em cada um dos arcos deste exemplo, o teste é especificado simplesmente por T (sempre verdadeiro). Mas este não precisa ser o caso. Suponha que, quando o primeiro SN for encontrado, seu número seja determinado e registrado em um registro chamado *número*. Depois os arcos marcados V poderiam executar um teste adicional que verificasse se o número encontrado daquele verbo em particular era igual ao valor armazenado em *número*. Testes mais sofisticados, envolvendo marcadores semânticos ou outros recursos semânticos, também podem ser realizados."[RIC 94]

4.9 Tratamento de Erros

É freqüente o usuário cometer erros, seja de significado, sintaxe, esquecimento e má interpretação. Antes de haver uma recusa para a resposta é tentador tentar adivinhar o que ele quis dizer. Na prática o melhor é encontrar a forma correta mais próxima, submetendo-a à confirmação pelo emissor.

No início dos anos 50, começaram aparecer os primeiros trabalhos de pesquisa relativo ao tratamento informatizado das línguas

naturais. As ferramentas de tratamento de erros em um texto permitiram, numa primeira etapa, o tratamento de erros a nível léxico.

O mercado atual é caracterizado por ferramentas que dominam o universo da palavra, enquanto que ainda não pudemos vivenciar uma evolução significativa a nível de estrutura da frase e de seu significado. Atualmente, vários são os programas comercialmente disponíveis no que se refere ao tratamento de erros léxicos, podemos citar WORD da Microsoft, MANUSCRIPT da Lotus, entre outros. O fator que desencoraja o usuário a utilizar revisores automáticos de texto, além do considerável tempo de resposta é que, certas ferramentas ou não detectam todas as palavras erradas presentes no texto, ou assinalam como erradas palavras corretas.

Alguns erros, como os erros de natureza fonética, não são suficientemente tratados, como é o caso de *hássido* que tem a grafia foneticamente equivalente a *ácido.* Erros causados pela utilização de desinência incorreta, como exemplo: *atora* dificilmente seria corrigido por atriz.

Do ponto de vista lingüística clássico, durante a análise de uma frase podemos encontrar erros distribuídos em três níveis diferentes: o nível léxico, o nível sintático e o nível semântico, ou seja, erros a nível de palavra, da construção e do sentido respectivamente.

A nível léxico, distinguem-se os erros de ortografia, os erros fonéticos, como a transposição de x por ss, e os erros de geração, como o uso de uma desinência incorreta do plural – calcãos. Os erros tipográficos, como a transposição das letras **e** e **m** em mesa (emsa) são igualmente incluídos nesta categoria.

A nível sintático, devem ser considerados os erros no que diz respeito a regras de construção de frases e a concordância entre seus componentes. Por exemplo, nas frases: “*índio comer salada***”** e **“** *Mostre clientes todos***”** podem ser consideradas como sintaticamente errônea, em

vista do erro de concordância. O reconhecimento de erros a nível sintático em um texto constituem um problema mais delicado. A verificação sintática exige à aplicabilidade de um conjunto de regras que modelam as relações entre os componentes. Entre as relações, encontram-se a ligação entre o nome e seu determinante, o nome e seu adjetivo, o sujeito e o verbo, bem como as conjunções, sejam elas coordenativas ou subordinativas.

A nível semântico, a complexidade e consequentemente a dificuldade de detecção de um erro aumenta. Ainda assim, os erros semânticos podem ser tratados. Por exemplo: “O pé da mesa está morrendo”. Consideramos que, o pé da mesa é um ser inanimado, sem vida, portanto, não pode então estar morrendo. Para tratar esse tipo de erro, deve-se especificar algumas características da palavra, ou seja, as especificações de cada palavra devem estar relacionadas e adicionadas na base de dados. Para a palavra “pé” por exemplo, sua especificação indicaria que é um ser inanimado, mas também, com vida, vegetal, etc. (possuiu variadas características por ser uma palavra que pode gerar ambigüidade), já no caso da palavra “mesa” indicaria que é um ser inanimado, madeira. Quanto a “morrendo”, sua especificação seria: ser vivo, humano, animal, vegetal etc. Testa-se então: não poderá um ser inanimado como é o caso de *mesa* concordar com o verbo *morrer* se este refere-se a seres com vida. A expressão “concerta-se pneu” deixa transparecer um erro difícil de tratar, observando que, se a expressão fosse “concerto de gaitas” não detectaríamos erro algum .

Os erros pragmáticos são provenientes do emprego inapropriado de uma frase, que por si só, não é errônea. Tais erros violam as leis do discurso, ou violam um cenário, um script. Por exemplo: *Antes do invento da máquina de escrever, o computador era a ferramenta mais usada.* A perspectiva de chegar-se a um sistema completo de tratamento de erros se apresenta como um longo caminho a percorrer.

4.9.1 Erros Comuns

- Erros de Ortografia

Não existe um processo de que seja infalível para tratar esse tipo de erro, mas pode-se utilizar alguns métodos para recuperar esta forma de erro. Uma palavra é composta de letras a *priori* imutáveis. Então, quando uma palavra é mal escrita, ela é comparada com as diferentes formas armazenadas no léxico, no intuito de encontrar a mais adequada, ou seja, a mais semelhante da solicitada.

- Falta de Informações

Um erro difícil de ser tratado ou até detectado é a falta de informação. Não dá a possibilidade de recuperação, pois a dificuldade esta em aonde e como assinalar as ausências. Outra dificuldade diz respeito a incoerência entre as informações, da mesma forma, pouco provável de ser detectada.

5 Sistemas desenvolvidos com interface de linguagem natural

Vários são os sistemas que vêm sendo desenvolvidos com interface de linguagem natural. Não deve surpreender então, que a tecnologia continue avançando e permitindo-nos construir ferramentas que aumentem nossa limitada, porem inigualável capacidade humana.

5.1 Baseball, Student e Eliza

Os primeiros programas em linguagem natural procuravam obter somente resultados limitados em domínios específicos, podemos citar: BASEBALL de Green, STUDENT de Bodrow e ELIZA de Weizenbaum. As sentenças de entrada eram simples interrogativas, do qual o programa retirava palavras-chaves que eram utilizadas para encontrar a resposta. Por causa do domínio restrito e do processamento de palavras-chaves, estes sistemas ignoram a complexidade da linguagem.

Os princípios usados pelo ELIZA são muito fáceis de descrever. Inicialmente, varre a entrada em busca de palavra-chave, que, quando detectadas, disparam sempre a mesma ação, que consiste em ou devolver uma mensagem padrão, ou usar parte da entrada para construir uma mensagem. Por exemplo, se o usuário digitasse:

Odeio sorvete

O programa detectaria a palavra- chave “odeio” e responderia.

Não é bom odiar

Observe que esta resposta era dada, independente do restante da frase. Se isso fosse tudo o que ELIZA fizesse, ficaria muito fácil descobrir que se tratava de um programa, pela quantidade limitada de respostas que daria. Entretanto, para introduzir alguma variação em suas resposta, o

programa fazia uso de sentença de entrada. Qualquer sentença que o usuário digitasse seria varrida em busca de certas palavras ou frases tais como “meu” é transformado em “seu” ou “você é” em “eu sou” . A finalidade destas transformações simples é devolver ao usuário as sentenças que tenha introduzido, como se estas fossem geradas pelo programa. Por exemplo, recebendo a sentença:

Você é um idiota

o computador devolveria

sou um idiota

possivelmente acompanhada de alguns sinais de exclamação ou de interrogação. Estas duas técnicas, resposta a palavras-chaves e alterações de tempos verbais, acompanhada de alguns outros truques especializados, podem produzir um programa que mantenha uma conversação razoável com o usuário.

Outra abordagem (PROTO-SYNTHEX –I) de Simmongs, Burger e Long – 1966, onde utilizavam memórias semânticas. Estes sistemas estocam a representação do texto numa base de dados, usando uma variedade de esquemas (indexação inteligente) para recuperar materiais contendo palavras e frases específicas. Respondem somente a perguntas para as quais foi explicitamente armazenada na base de dados, ou seja, não utilizam nenhum mecanismo de inferência.

5.2 Prospector

PROSPECTOR foi desenvolvido nos fins da década de 70 no Stanford Research Institute, por um grupo que incluía Peter Hart, Richard Duda, R. Reboh, K. Konolige, P. Barret e M. Einandi. O desenvolvimento do PROSPECTOR foi financiado pelo U. S. Geological Survey e pela National Science Foundation.

O PROSPECTOR destina-se a oferecer consultas aos geólogos nos primeiros estágios da pesquisa de um local sobre depósitos de teor

mineral. Os dados são sobretudo observações geológicas da superfície e partem do princípio de que são incertos e incompletos. O que difere o PROSPECTOR de outros sistemas especialistas é que, emprega uma interface de linguagem natural restrita (LIFER) permitindo aos usuários digitarem frases da mesma forma que fariam perguntas a um consultor de geologia.

5.3 Rendezvous e Intellect

O RENDEZVOUS, um sistema experimental desenvolvido pela Codd na IBM Research, e o INTELLECT, originalmente desenvolvido por Harris em Dartmouth College (sob o nome de ROBOT) agora disponível através de Artificial Intelligence Corporation (AIC) e também pela IBM sob licença da AIC (entretanto nem sempre sob o nome de INTELLECT), são sistemas mais conhecidos em linguagem natural.

RENDEZVOUS opera transformando a consulta original em uma expressão de cálculo relacional, ao fazê-lo usa uma base de conhecimento. A transformação da consulta original é feita de modo fragmentário. De modo geral, poderá haver fragmentos que o RENDEZVOUS não entenda. Para tanto ele utiliza um diálogo de esclarecimento com o usuário, tentando extrair os fragmentos que não foram entendidos.

A estrutura interna do INTELLECT é bastante diferente. Primeiro este sistema não oferece diálogo de esclarecimento com o usuário, emprega o conhecimento embutido da sintaxe inglesa. Se a consulta oferece possibilidades de várias interpretações, ele pesquisa o banco de dados na tentativa de descobrir os significado pretendido pelo usuário. Para isto, ele contém uma extensão ampla e dinâmica do dicionário. O que o INTELLECT utiliza para a confirmação é “repetir” a consulta para o usuário com o intuito de esclarecer qual é exatamente a consulta que o sistema está respondendo.

A idéia do INTELLECT parece ser preferível ao do RENDEZVOUS, pois envolve menos esforço por parte do usuário. De outro lado, a pesquisa ao banco de dados poderá ser cara se o banco de dados for grande. Além disso RENDEZVOUS é mais apropriado quando a consulta é realizada distante fisicamente do sistema central, porque não há necessidade de acessar o banco de dados (exceto quanto ao catálogo que pode estar localmente) até o momento da versão final e completa da consulta.[HAR 88]

Todos os programas acima citados são somente de leitura, operações de atualização não podem ser feitas. A entrada em linguagem natural é feita de forma restrita e digitada. O produto INTELLECT, serve como front-end para a linguagem FOCUS (4GL).

Criar um razoável sistema de Linguagem natural é um processo lento e caro. O INTELLECT exigiu centenas de homem/ano de trabalho para ser desenvolvido, até o ponto em que usuários finais não treinados pudessem usá-lo com confiança.

Temos ainda o THEMIS e o MPG ENGLISH, suas funções são de aceitar queries em inglês natural e traduzi-los em uma série de comandos que controlam a interação de quais sejam as funções do sistema necessárias para produzir uma resposta. Existem ainda muitos sistemas em teste e outros que se utilizam da linguagem query, ou seja, linguagem de pesquisa, mas de forma muito primitiva.

6 Ambientes e Instrumentos

Neste capítulo, daremos uma visão das características gerais das linguagens e dos diferentes ambientes e instrumentos empregados para desenvolver sistemas cognitivos.

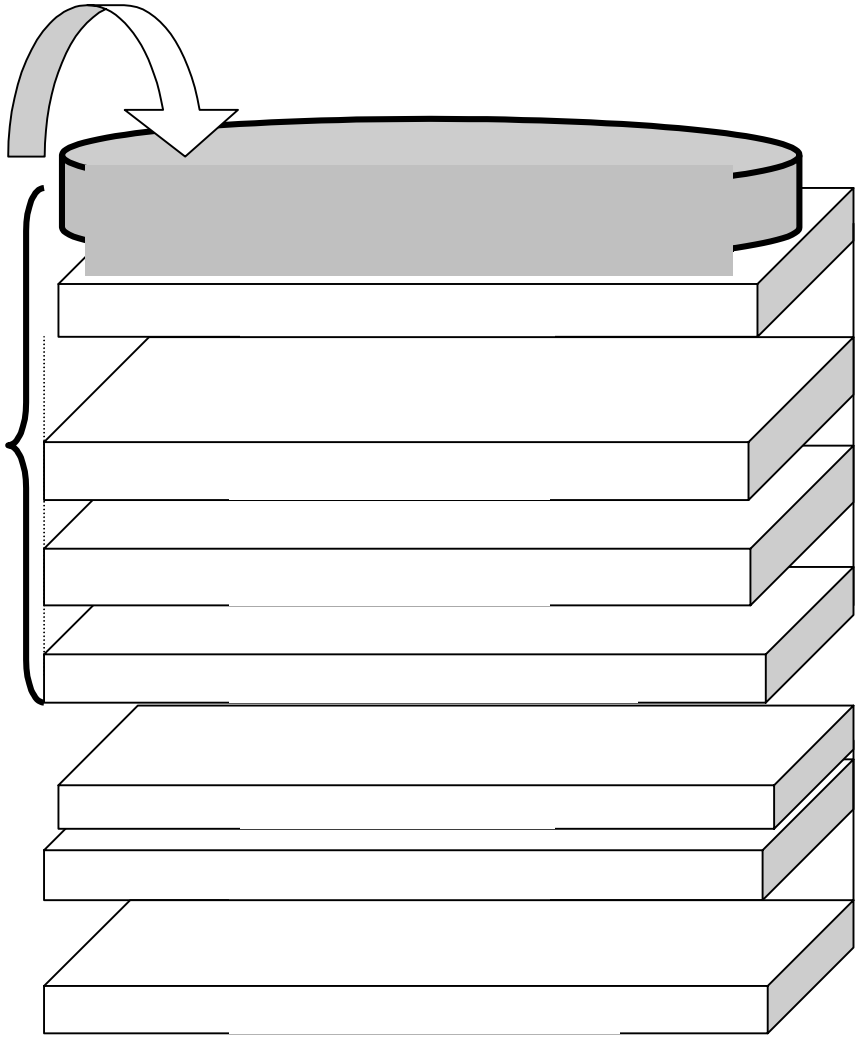

Figura 6.1 - Os níveis da programática entre os problemas humanos e o equipamento
Fonte - HARMON e KING - Sistemas Especialistas

A figura 6.1 oferece uma panorâmica dos vários níveis de programas que existem entre uma situação humana de problema e o

equipamento do computador. No cimo da figura, estão os problemas, se decidimos construir um sistema cognitivo para ajudar o usuário, é instrutivo considerar exatamente quais os níveis que jazem sob o sistema cognitivo. Devemos considerar o equipamento, a linguagem, o ambiente, o instrumento, o conhecimento e o problema que se quer resolver. [HAR 88]

6.1 Linguagens

Para definir qual a melhor linguagem de programação a ser usada ao implementarmos um sistema de interface natural, devemos analisar todo o cenário existente. O que queremos desenvolver, quais são as restrições, as necessidades e principalmente quais objetivos queremos alcançar. Qualquer que seja o processo, ele sempre existirá num ambiente de desenvolvimento, que é a soma total de hardware e software. Pode-se fazer muitos trabalhos interessantes em ambientes relativamente simples, mas a nova geração de software e hardware nos impele a examinar de onde viemos e o que será possível e necessário no futuro.

Atualmente, os programadores COBOL sentam-se em seus terminais e utilizam um editor de texto para elaborar o código-fonte ou corrigir os erros da última compilação do sistema que estão construindo. Feitas as alterações, eles saem do programa editor e invocam o compilador COBOL e o editor de linkagem. Aguardam, pela execução do job de compilação do programa. Se o programa for compilado e linkado sem erros, o programador submeterá um conjunto de comando JCL para testar o programa, esperando novamente para o job ser executado. Este ciclo se repete muitas vezes até que o programa funcione conforme especificações. Ainda, durante o processo, o programador deve estar ciente de cada detalhe da operação do programa: tamanho de memória, definições de variáveis, condições de *looping* e muitos outros. Este cenário descreve o ambiente de desenvolvimento, um tanto penoso, no qual os programadores de COBOL trabalham atualmente.

Os programadores de IA que usam máquinas LISP têm uma experiência bastante diferente. Em um ambiente LISP pode não haver compilador e o editor de programas é uma parte integrante do mesmo sistema que executa o código. O programa é escrito e depurado numa única série de ações integradas. Se o programa funcionar, ele então será submetido a um compilador uma única vez, após o que será usado num ambiente de produção. Além disso, podem existir recursos inteligentes de apoio ao programador responsáveis pelo tratamento da verificação e por toda a documentação que seria necessária num programa COBOL. Os assuntos relacionados com tamanho de tabelas, tipos de variáveis e outros afins são, agora, funções para o ambiente de desenvolvimento de IA.

O ambiente contém protótipos depurados de tipos de programas freqüentemente usados, de tal forma que o programador precise somente fazer alterações específicas em seu sistema. Ainda, o sistema mantém um histórico das entradas efetuadas pelo usuário e seus efeitos, as quais podem ser modificadas ou repetidas, sendo possível excluir uma série de entradas, isto tudo, enquanto o sistema testa o programa.

Tentar implementar um sistema baseado em COBOL, mesmo que possível, não é nada fácil ou divertido, reduzindo ainda a experiência de aprendizado. Outras linguagens oferecem um ambiente melhor, como BASIC, preferencialmente PASCAL ou C e, na melhor das opções, LISP ou PROLOG (LISP- Linguagem de processamento de listas). Ainda, caso deseja-se trabalhar com acessos a base de dados em ambientes de 4ª geração, pode-se optar por trabalhar diretamente com a própria linguagem do banco de dados.

As linguagens da IA têm características embutidas que tornam mais fácil de construir sistemas especialistas ou sistemas com interface em linguagem natural. Destina-se a manipular, por exemplo, o processamento simbólico. As linguagens convencionais destinam-se essencialmente a manipular operações numéricas. É muito mais prático programar um

instrumento ou um sistema cognitivo em uma linguagem de IA do que usar uma linguagem convencional.

Entretanto, o LISP e o PROLOG são menos conhecidos que as linguagens convencionais (FORTRAM, C, PASCAL, 4GL, etc.). Em relação ao uso da linguagem LISP, deparamos com um outro problema, rodam lentos, uma vez que, os sistemas operacionais convencionais “traduzem” o LISP de maneira ineficaz para a linguagem de máquina. Durante anos os pesquisadores em IA procuraram desenvolver um equipamento que usasse o LISP como sistema operacional. Estas máquinas, chamadas Máquinas LISP, rodam sistemas cognitivos muito mais rápida e eficientemente que os equipamentos convencionais que usam um sistema operacional padrão.

A programação lógica existe desde o início dos anos 70 (LISP), mas não teve sucesso até que a linguagem PROLOG apareceu. PROLOG e suas possíveis linguagens derivadas, ainda não inventadas, provavelmente terão um impacto importante no uso tradicional da análise estruturada. Ela introduz uma nova maneira de pensar no desenvolvimento de sistemas sem descartar as boas características do processo tradicional. PROLOG é uma linguagem de especificação descritiva. Ela resolve uma ampla classe de problemas. É parte componente da Quinta geração de linguagens.

Por outro lado, temos as linguagens de Quarta geração (4GL’s), elas são adequadas a uma classe limitada de problemas, nos quais processos simples e bem determinados são aplicados a base de dados para produzir respostas bem definidas. Além disso, são restritivas quanto à maneira que a informação é armazenada. A principal vantagem é que, a maioria dos ambientes de trabalho possuem pelo menos uma a disposição, e podem ser utilizadas na produção de protótipos de modo relativamente fácil.

Na verdade, podemos construir sistemas independente do ambiente, o que conta é se ele é realmente adequado e eficaz ao que se quer atingir. Considera-se a virtude de usar o LISP ou o PROLOG pela

possibilidade de desenvolver instrumentos mais complexos ou refinados e visarem mais o futuro imediato que o presente.[HAR 88]

7 Modelo do Protótipo

Consideramos no decorrer do trabalho os níveis lingüísticos e os modelos de gramáticas existentes para a construção de um sistema de interface natural. Então, abordaremos neste capítulo os aspectos operativos, os quais basearam-se no referencial acima.

Nas fases de processamentos da linguagem encontramos geralmente :

- analisador morfológico e lexical
  é um problema resolvido, com tempo de processamento dependente do tamanho do dicionário utilizado. A decisão de qual o tipo de dicionário utilizar depende dos objetivos do sistema e do espaço disponível.

- analisador sintático e semântico
  uma mesma frase pode ter mais de uma estrutura derivável, denotando interpretações diferentes. A resolução desta ambigüidade não é possível a nível sintático , devendo este nível gerar as interpretações necessárias e serem tratadas a nível semântico.

- Analisador pragmático
  Um aspecto que merece um estudo bem mais aprofundado é realizar o processamento a nível pragmático. Em certas frases o efeito pretendido é claramente declarativo, esta é precisamente a atitude correta. Mas, em outras frases, o efeito pretendido é diferente. Podemos descobrir este efeito aplicando um conjunto de regras. A pragmática trata da coerência do texto, como resultado de processos cognitivos que se operam entre os interlocutores.

Devido a grande complexidade que denotam os processadores semânticos e principalmente ao longo e indefinido período de tempo necessário para mapear praticamente todas as possibilidades é que definimos não ser possível fazer uso de um analisador semântico. Para tanto, desenvolvemos um modelo baseado em gramáticas puramente sintáticas.

Para a implementação utilizou-se um analisador léxico, juntamente com as classes gramaticais e os comandos existentes do SQL, formando assim, um dicionário de dados. Foi implementado também um analisador sintático, o qual trata a sintaxe da frase.

Quanto ao ambiente optou-se pelo Banco de Dados ZIM, pelo fato de ter sido ministrado nas disciplinas de Banco de Dados e estar atualmente disponível na universidade. Foi considerado também a característica de portabilidade de sistemas operacionais, ou seja, compatível com mais de uma plataforma e principalmente por ser o SGBD em que temos maior conhecimento.

Para melhor compreensão do problema e para facilitar a definição do protótipo, a figura 6.1 mostra a representação das estruturas utilizada na implementação:

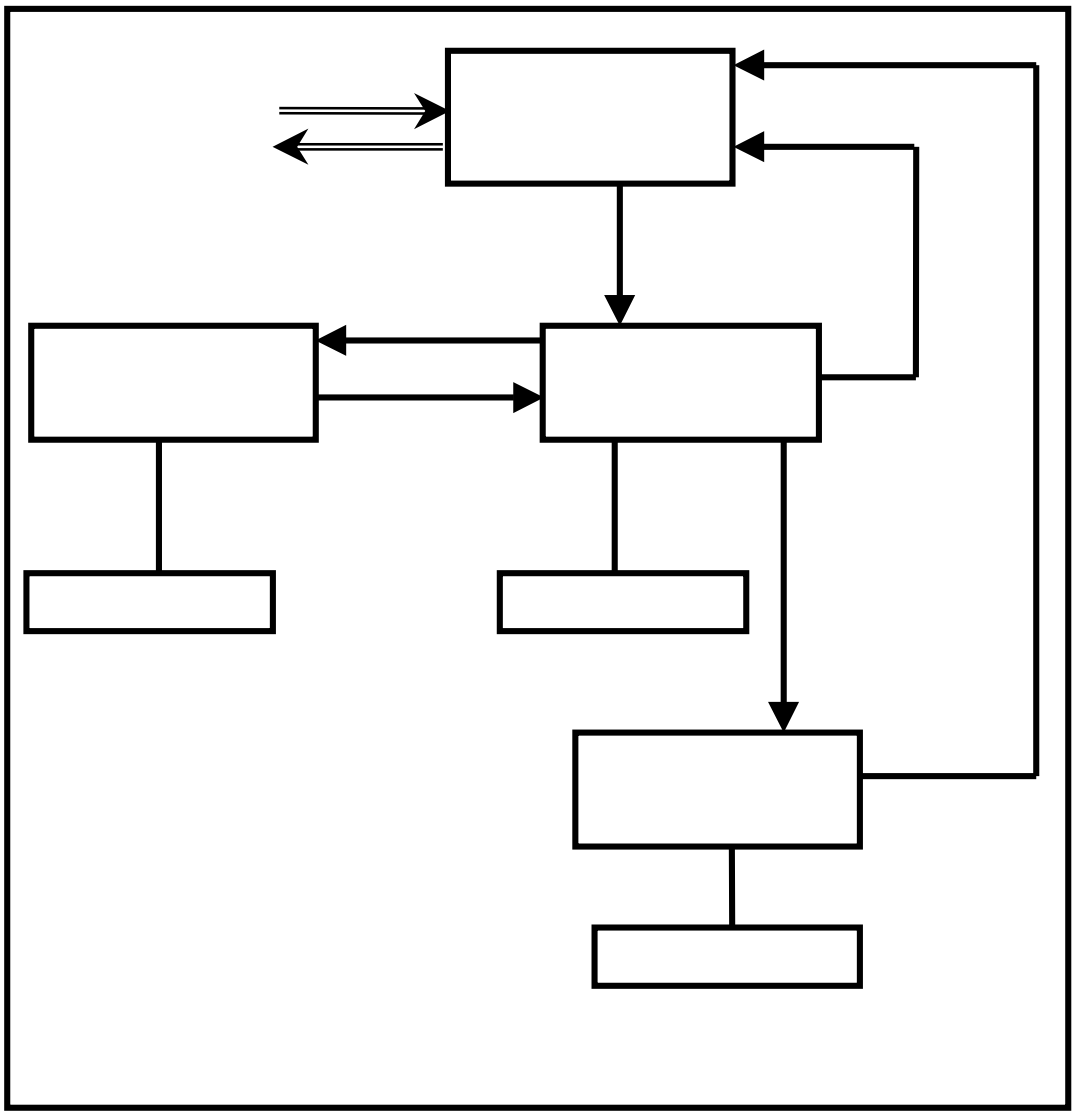

Figura 7.1 – Esquema do protótipo

Interface: é responsável pela comunicação direta com o usuário, ou seja, ler as sentenças por ele digitadas e mostrar-lhe os resultados. Também compete à interface auxiliar o usuário quanto a operação do sistema.

Analisador léxico: verifica a existência de cada palavra da sentença e a classifica segundo sua classe (verbo, substantivo, artigo, etc.). Classifica também, cada palavra da sentença em uma correspondente em SQL, se está existir.

Analisador Sintático: é responsável pela verificação sintática da sentença enviada, ou seja, verifica se a sentença pertence à linguagem definida pela gramática. Se responder com sucesso passa para o analisador SQL, ao contrário retorna ao usuário uma mensagem de erro.

Analisador SQL: Recebe a frase do analisador sintático e verifica se a mesma possui palavras que correspondem a comandos do SQL, já classificados no léxico. Se positivo, gera a sentença em SQL e a executa.

Gramática: Estão armazenadas todas as regras que determinam a estrutura da linguagem e a função dos elementos desta estrutura. É a partir da gramática que podemos, por exemplo, saber que a frase " Liste os Frederico Westphalen" está errada.

Dicionário: Armazena as palavras e seus sinônimos, sua categoria e sua função SQL, se corresponder a um comando.

7.1 Analisador Léxico

O analisador léxico verifica caracter por caracter até encontrar um espaço em branco, considera então uma palavra. Verifica se esta palavra existe no léxico, se positivo, verifica sua classe gramatical correspondentes. Ainda no léxico testa se a palavra corresponde a algum comando SQL.

7.1.1 Léxico (dicionário)

No léxico encontra-se a taxinomia das palavras. Exemplificados resumidamente em:

Determinante (artigo)

determinante (o,a)

determinante (os,as)
determinante (um, uma)
determinante (uns, umas)

Pronomes

interrogativo (quantos)
relativo (todos)

Preposição

preposição (a)
preposição (com)
preposição (de)
preposição (de+a)
preposição (em)

Conjunção

conjunção (e)
conjunção (que)

Advérbio

Lugar (onde)
Modo (como)
Intensidade (mais, menos)

Verbo

Verbo (liste, mostre, relacione)
Verbo (listar, mostrar, relacionar)

Substantivo

comum (cliente)
comum (clientes)

Ainda no léxico se encontram as palavras que correspondem a valores de campo (com respectivos sinônimos se houverem) da base de dados Clientes,

exemplo:

- Nome
- Endereço
- Bairro
- Cidade
- Cônjuge (esposo, esposa)
- Salário (renda)
- Sexo

Se para uma palavra existente no dicionário existir um comando SQL, esta também constará no léxico. Exemplo:

Liste	⟶	Select
Relacione	⟶	Select
Onde	⟶	Where
Com	⟶	Where
Todos	⟶	*
Igual	⟶	=
=	⟶	=
Maior	⟶	>

O dicionário de dados do protótipo é compostos por:

Tabelas/Entidades

- Categoria
- cliente
- coluna
- dicionário
- funcaosql
- gramática

Campo	Tabela	Tipo	Tamanho	Indexado
cacodcategoria	categoria	char	5	unique
cadescricao	categoria	varchar	20	unique

Campo	Tabela	Tipo	Tamanho	ndexado
clcodcli	cliente	int	4	unique
clnome	cliente	varchar	30	yes
clendereco	cliente	varchar	30	no
clbairro	cliente	varchar	20	yes
clcep	cliente	char	8	no
clcidade	cliente	varchar	30	yes
cldatanascimento	cliente	date	8	yes
clfone	cliente	char	11	no
clsexo	cliente	char	1	yes
clestadocivil	cliente	char	10	yes
clconjuge	cliente	varchar	30	no
clsalario	cliente	vastint	11	yes

Campo	Tabela	Tipo	Tamanho	Indexado
codescricao	coluna	varchar	20	unique
cotipo	coluna	char	10	yes

Campo	Tabela	Tipo	Tamanho	Indexado
dipalavra	dicionário	varchar	20	unique
dicodcategoria	dicionário	char	5	yes
dicodfuncaosql	dicionário	char	2	yes

Campo	Tabela	Tipo	Tamanho	Indexado
fucodfuncaosql	funcaosql	char	2	unique
fudescricao	funcaosql	varchar	20	unique

Campo	Tabela	Tipo	Tamanho	Indexado
grcodgra	gramática	char	5	yes
grcodcategoria	gramática	char	5	yes
grproximo	gramática	char	5	yes

7.2 Analisador Sintático

No protótipo o Analisador Sintático tem a função de analisar e converter a frase digitada para ser avaliada pelo analisador SQL. Para atingir este fim, ele interage com o léxico e a gramática. Com o primeiro para

fornecer a classificação das palavras, e com o segundo para verificar a função sintática desta palavra na frase.

7.2.1 Regras determinísticas e indeterminísticas

O analisador sintático deverá “provar” regras, tanto determinísticas (com uma hipótese de solução) como indeterminísticas (com n hipóteses de solução).

As regras são montadas com dois componentes: lado esquerdo e lado direito.

Ex: V → verbo A
Leia-se: um V é um verbo seguido de A

O lado esquerdo da regra é o que deseja provar, o lado direito as condições para o lado esquerdo ser verdadeiro.

O lado direito pode combinar elementos terminais e não-terminais, desta forma todos os elementos devem ser provados para que a sentença seja verdadeira.

Quando ocorre indeterminismo o analisador deverá provar uma das regras definidas, caso não consiga, tenta provar a outra, e assim sucessivamente até conseguir provar uma das regras, ou esgotarem as hipóteses. Neste caso a frase não pertence a linguagem definida pela gramática.

Ex:
B → substantivo
B → substantivo C
Leia-se: um B é um substantivo ou um substantivo seguido de C.

O indeterminismo aumenta consideravelmente o tempo que o Analisador Sintático dispende para analisar a frase. Por exemplo:

a) B → substantivo
b) B → substantivo C
c) B → código C

Para se provar um B, ou se prova a primeira regra , ou a Segunda, ou ainda, a terceira. Quando tentar provar (a) se verifica se a palavra atual é um substantivo, ou seja, pedirá ao léxico para consultar a taxinomia. Caso não se trate de um substantivo a regra falha. Tentará provar (b) novamente terá consultado o léxico realizando uma tarefa já feita e falha. Tentará provar (c) novamente consulta o léxico agora para ver se a palavra é um código, o que resulta em sucesso. Poderá se tornar um processo lento e caro a medida que a gramática aumenta, isto é, adaptando casos de indeterminismos.

Exemplo 1
Relacione todos clientes
Verbo→ pronome → substantivo
Exemplo 2
Relacione todos clientes da cidade de Seberi
Verbo → pronome → substantivo →preposição → campo → preposição→valor

S	→	V
V	→	verbo A
A	→	pronome B
B	→	substantivo
B	→	substantivo C
C	→	preposição D
D	→	campo E
E	→	preposição X
X	→	valor

Figura 7.2 – Exemplo de gramática sintática

Para o exemplo 1 lê-se:

S é um verbo seguido de A
A é um pronome seguido de B
B é um substantivo

Para o exemplo 2 lê-se:

S é um verbo seguido de A
A é um pronome seguido de B
B é um substantivo seguido de C
C é uma preposição seguido de D
D é um valor de campo da base de dados seguido de E
E é uma preposição seguida de X
X é um valor de registro na base de dados

7.2.2 Gramática

Uma gramática realiza testes para que frases que não tenham uma sintaxe correta não sejam aceitas pelo analisador.

Gramática:

S	→	ver A
A	→	art B
A	→	pronr B
A	→	pronr C
AP	→	pr I
AV	→	ver2 AP
B	→	subc
B	→	99 D

B → subc K
B → subc E
B → 99 A
B → subc R
C → art B
D → pr B
E → adv F
E → adv O
E → con F
F → 99 G
G → adv H
G → adv I
G → ver2 I
H → art I
I → valor
I → valor J
I → valor AV
J → con F
K → pr L
L → 99 M
M → pr I
O → art F
S → ver A
R → pron T
T → 99 M
F → subc2 F
B → subc2 D
A → verax A
F → 99 FS
FS → subc G

Onde:

99 - código referindo-se a valor de campo

valor - refere-se ao registro do campo 99

Exemplos de sentenças que são aceitas pela gramática acima:

- Mostre todos os clientes
- Liste os clientes
- Relacione o nome, o endereço dos clientes
- Liste o nome, o salário dos clientes onde sexo for F
- Relacione os clientes com salário igual a 500
- Quero os clientes da cidade de Frederico
- Mostre os clientes que residem em Frederico
- Liste o nome dos clientes que residem em Seberi

A gramática utilizada no protótipo é uma gramática com processamento sintático. Na sua especificação foram utilizadas abreviaturas (alfabeto) para o conjunto dos não-terminais e um código para referenciar ao valor de campo da base de dados Cliente.

Se a gramática da figura 6.3 fosse processada para a seguinte sentença de entrada:

Liste todos os clientes da cidade de Frederico

liste	=	verbo (ver)
todos	=	pronome (pronr)
os	=	artigo (art)
clientes	=	substantivo (subsc)

As regras para criação da árvore de derivação são:

1) Comece no estado S que é um verbo, seguido de A

S → ver A

testa se a palavra **liste** é verbo. Se for positivo, vá para o estado A

2) Estando no estado A teremos ramificações:

A → art B
A → pron B
A → pron C
A → verax A

Testa se a palavra **todos** é um artigo (A→art B), este teste falha.
Testa então se **todos** é um pronome (A→pron B). Se sucesso, vá para o estado B.

3) Estando no estado B, teremos:

B → subsc
B → 99 D
B → subc E
B → subc K
B → 99 A
B → subc R
B → subc D

Testa se a palavra **os** é susbtantivo (B→subsc), este teste falha, vá para o próximo B.
Testa então se **os** é um valor de campo (B→99 D) , falha novamente.
Passa por todos os "Bs", testando se é um artigo. Falham todos os testes.
Retorna então ao estado A.

3)Estando no estado A, vá até o nó posterior ao último percorrido (A→pron C).
Testa novamente a categoria, se a palavra **todos** é pronome, sucesso, então segue até C.

4) Estando no estado C, testa se a palavra **os** é artigo (C→ art B), este teste é positivo então segue até B

5) Estando no estado B

B → subsc
B → 99 D
B → subc E
B → subc K
B → 99 A
B → subc R
B → subc D

testa se a palavra **Clientes** é substantivo seguido de um valor nulo (B→subc), se positivo, então fim.

7.3 Analisador SQL

Os comandos SQL utilizados pelo protótipo estão armazenados no léxico, ou seja, se para um valor no léxico existir um comando SQL este o armazenará. O analisador SQL verificará então, se a sentença de entrada formou uma sentença em SQL e se a mesma está correta. Se o resultado é positivo então o analisador manda executar.

Para conseguir executar um comando em SQL padronizou-se que a sentença de entrada quando se referir a um registro da base de dados deverá conter o nome do campo ou um sinônimo da palavra que se quer referenciar. Por exemplo:

a) Sentenças válidas:

- Relacione o nome dos clientes da cidade de Iraí
- Relacione os clientes que moram em Iraí".
- Liste o cliente onde a renda mensal for superior 1000

b) Sentenças inválidas:

- Relacione o nome dos clientes de Iraí
- Relacione os clientes de Iraí
- Liste o cliente onde for superior 1000

No item b, a primeira e a segunda sentenças são inválidas. Não foi informado que “Iraí” é cidade ou algum sinônimo para a palavra cidade, como morar, residir. O mesmo ocorre na terceira sentença, onde também não foi informado que “superior a 1000” é salário ou um sinônimo, como renda, rendimentos, etc.

8 Conclusão

Este trabalho apresentou uma visão geral das etapas que envolvem o reconhecimento e processamento da linguagem natural na computação. Descrevemos os conceitos envolvidos no tratamento computacional, em nível léxico-morfológico, sintático, semântico, pragmático e do discurso. Problemas inerentes ao processamento da linguagem foram levantados, podendo ser objeto de estudo para trabalhos futuros.

Todas as linguagens são geradas por um conjunto de regras. Mesmo assim no universo das línguas naturais, a definição de gramáticas sensíveis aos fenômenos observados ainda não foi alcançada. Pode-se até mesmo questionar a exeqüibilidade de definição de uma gramática exaustiva para as línguas naturais.

O desenvolvimento de sistemas, ainda que restritos, de linguagem natural é indubitavelmente uma tarefa muito complexa. É um desafio, desenvolver sistemas, economicamente viáveis e em tempo aceitável, atendendo a critérios de qualidade.

Os progressos, no entanto, se ainda deixam a desejar do ponto de vista dos sistemas que sejam suficientemente gerais e poderosos para resolver problemas, ou suficientemente capazes de aprender em áreas especializadas, têm a vantagem de ensinar muito em termos de generalização e de pesquisa, e também possibilitar a aproximação do usuário com a máquina através de uma interface amigável.

Diante da complexidade, o resultado obtido em relação ao protótipo foi positivo, proporcionando-nos uma série de questionamentos e respostas, apesar de não encontrarmos bibliografia que referencia acesso a base de dados, criamos um modelo, no qual a sentença em linguagem natural é convertida em comandos SQL, gerando um arquivo. O Banco de dados executa a sentença gerada e apresenta o resultado para o usuário.

Embora a experiência tenha sido realizada em domínio específico, com vocabulário restrito e de tratamento apenas sintático, ficou evidente que o desenvolvimento em linguagem natural para domínio restrito é viável.

É importante porém, que para um trabalho de continuidade sejam introduzidas melhorias. A implementação deve ser mais modular e genérica. Dar uma continuidade ao tratamento sintático e implementar o semântico, é apenas uma questão de tempo.

Convém ressaltar que a interdisciplinaridade, principalmente na área de lingüística, é um pré-requisito para que bons trabalhos possam ser realizados.

Muitos são os problemas a serem resolvidos para que se possa simular em máquina o comportamento inteligente das pessoas. A perspectiva de chegarmos a um sistema completo no processamento da linguagem natural ainda se apresenta como um longo caminho a percorrer. No entanto, com o resultado deste trabalho e principalmente com o da implementação do protótipo, podemos ver que é possível dar a este uma interface na linguagem dos usuários.

Bibliografia

[AND 79] ANDRÉ, Hildebrando A. de . **Gramática Ilustrada**. Editora Moderna-São Paulo, 1979.

[AGU 95] AGUSTINI, Alexandre. **Estudo Inicial sobre o Processamento da Linguagem Natural** – Trabalho Individual I – PUC – Curso de Mestrado em Informática – 1995.

[COU 92] COULON, Daniel e KAYSER, Daniel. **Informática e Linguagem Natural**. Brasília: IBICT ; Rio de Janeiro: SENAI – 1992.

[CHO 71] CHOMSKI, Noam. **Linguagem e Pensamento**. 2. Ed. – Vozes – Petrópolis, 1971.

[DAT 89] DATE, C.J.. **Guia para o padrão SQL**. Ed. Campus-Rio de Janeiro, 1989.

[HAR 88] HARMON, Paul e KING, David. **Sistemas Especialistas**. Editora Campus Ltda – Rio de Janeiro, 1988.

[HUN 87] ROBIN, Hunter. **Compiladores**. Editora Presença Ltda – Impr./Acabamento. Artes Gráficas Ltda. – Marfa, 1987.

[KEL 91] KELLER, Roberto. **Tecnologias de Sistemas Especialistas**. Makron Books – São Paulo, 1991.

[KOW 83] KOWALTOWSKI, Tomasz. **Implementação de Linguagens de Programação**, Editora Guanabara Dois S/A – Rio de Janeiro, 1983.

[KOR 93] KORTH, Henry F. e SILBERSCHATZ, Abraham. Sistemas de Banco de Dados. 2.ed. Makron Books - São Paulo, 1993.

[LIM 96] LIMA, Vera Lúcia Strube de. **Processamento da Linguagem Natural – premissas e desafios**. Anais – IV Escola Regional de Informática, 1996.

[LOB 86] LOBATO, Lúcia Maria Pinheiro. **Sintaxe Gerativa do Português**. Editora Vigíla Ltda – Belo Horizonte, 1986.

[PER 76] PERINI, Mário A. . **A Gramática Gerativa**. Editora Vigília Ltda – Belo Horizonte, 1976.

[PUC 90] PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL. **Letras de Hoje**. Conceitos de Inteligência Artificial aplicados a Lingüística – Bertilo Frederico Becker - pg. 15-40. 1990.

[PUC 90] PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL. **Letras de Hoje**. Tratamento Automatizado da Língua Natural – Vera Lúcia Strube de Lima – pg. 41 a 54. 1990.

[RAP 83] RAPOSO, Eduardo Paiva. **Introdução à Gramática Generativa**. 2.ed.Moraes Editores- Rio de Janeiro, 1983.

[RIC 93] RICH, Elaine e KNIGHT, Kevin. **Inteligência Artificial**. Makron Books do Brasil - São Paulo,1993.

[SUE 97] SUERETH, Russell. **Developing Natural Language Interfaces**. McGraw-Hill – United States of America, 1997.