Processamento de Linguagem Natural

Flávia de Almeida Barros
Jacques Robin

UFPE - Universidade Federal de Pernambuco
CIn - Centro de Informática
Cx. Postal 7851 – CEP 50732-970 Recife (PE)
{fab,jr}@cin.ufpe.br

Resumo: O Processamento de Linguagem Natural (PLN) é um ramo da Inteligência Artificial que tem por objetivo interpretar e gerar textos em uma língua natural (*e.g.*, Português, Inglês, Francês, Espanhol, etc). Este texto apresenta inicialmente uma arquitetura genérica de sistemas para PLN, e algumas noções preliminares de Lingüística. Em seguida, são abordadas as duas sub-áreas de trabalho em PLN: interpretação e geração de texto. Este tutorial foi apresentado na Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação de 1996, e foi revisado em 1997.

Palavras Chaves: Inteligência Artificial, Processamento de Linguagem Natural.

Conteúdo

1 Introdução

O Processamento de Linguagem Natural (PLN)[1] é um ramo da Inteligência Artificial (IA) que tem por objetivo interpretar e gerar textos em uma língua natural (*e.g.*, Português, Inglês, Francês, Espanhol, etc).

O PLN é genuinamente multi-disciplinar, congregando, principalmente, estudos nas áreas de Ciência da Computação, Lingüística e Ciências Cognitivas. A pesquisa em PLN divide-se em duas sub-áreas de trabalho: interpretação e geração.

Interpretação de linguagem natural baseia-se em mecanismos que tentam 'compreender' frases em alguma LN, buscando traduzi-las para uma representação que possa ser compreendida e utilizada pelo computador (*e.g.*, a Lógica de Predicados).

Um exemplos clássico de aplicação nesta área são as interfaces em LN para Bancos de Dados (BDs). Esses sistemas traduzem a pergunta do usuário, em alguma LN, para a linguagem de consulta (query language) utilizada pelo BD em questão. As interfaces em LN tornam os BDs acessíveis a usuários leigos, para quem o aprendizado de uma 'query language' seria muito árduo.

Na *geração de linguagem natural*, o oposto se verifica: o computador traduz uma representação interna de um conteúdo semântico pré-definido para sua expressão em alguma língua. Esta sub-área de pesquisa busca produzir textos o mais próximos possível de textos produzidos por pessoas. Um exemplo clássico de aplicação nesta área é a geração automática de resumos textuais de dados armazenados em tabelas de grande tamanho.

Uma aplicação que reúne técnicas das duas áreas de PLN é, por exemplo, a tradução automática. Aqui, um texto em alguma língua natural (*e.g.*, Português) é traduzido para uma representação intermediária – *interpretação* – que em seguida é traduzida para um texto em outra língua natural (*e.g.*, Inglês) – *geração*.

Apresentamos abaixo a arquitetura genérica de sistemas para PLN, seguindo com algumas noções preliminares de Lingüística. A partir daí, veremos três aspectos da Interpretação de Linguagem Natural: mecanismos para o processamento sintático de frases, formalismos para a interpretação semântica de LN, e processamento automático do discurso.

2 Arquitetura de Sistemas para PLN

Sistema para PLN são, em geral, modulares, onde diferentes níveis de processamento são executados em módulos distintos (*cf.* Fig. 1). Esses módulos se comunicam pela passagem de representações intermediárias do texto sob análise. Apenas o fluxo de informação muda, de acordo com a tarefa do sistema – se interpretação ou geração.

Nos sistemas para interpretação de LN, temos o texto como entrada e uma representação formal como saída. Essa representação é dependente da aplicação – por exemplo: em interfaces para BDs, a saída é uma consulta em uma *query language*, e em um tradutor automático, a saída é uma representação conceitual do texto independente das línguas de origem e destino.[2] Na geração, o fluxo se inverte, e o texto é gerado a partir de uma representação formal do seu conteúdo esperado e dos seus objetivos de comunicação.

Sistemas para PLN utilizam Bases de Conhecimento (BCs), que são arquivos externos onde informações necessárias ao processamento dos textos são codificadas declarativamente. Na figura 1,

[1] Também denominado de Processamento de Língua Natural. Adotamos aqui o termo 'Linguagem Natural' por ser este mais difundido e tradicionalmente usado no Brasil.

[2] Esta representação intermediária é chamada de *interlíngua*.

podemos identificar cinco bases de conhecimento, representadas por figuras elípticas: a *gramática*, o *léxico* e o *modelo do discurso*, que contêm informação lingüística; o *modelo do domínio*, com informações sobre o domínio específico da aplicação, e o *modelo do usuário*, com dados sobre o usuário do sistema.

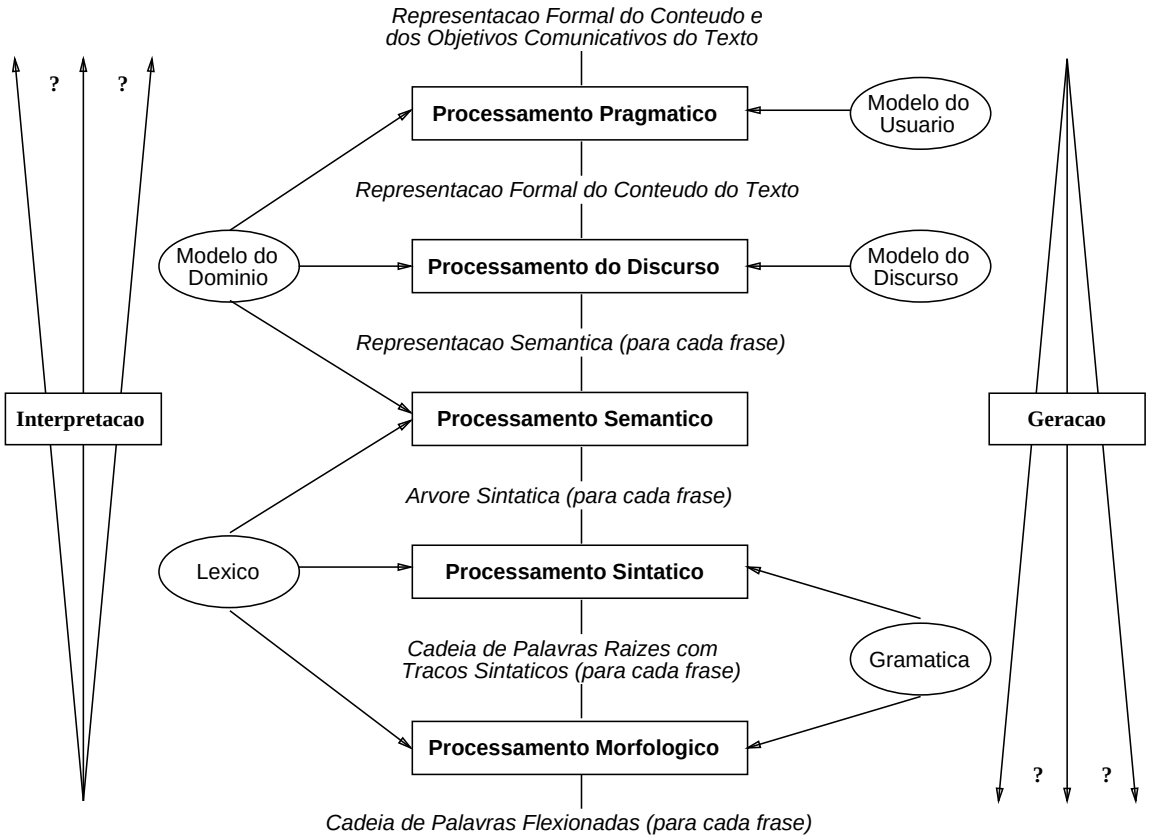

Figura 1: Níveis de processamento em PLN.

2.1 Bases de Conhecimento

2.1.1 O Léxico

Sistemas para PLN utilizam léxicos, que consiste em dicionários com os termos utilizados pelo sistema no processamento dos textos. Cada termo (ou palavra) no léxico pode estar associado às suas características:

- morfológicas (conjugação dos verbos, inflexão dos substantivos e adjetivos, etc)
- sintáticas (categoria gramatical, transitividade e regência verbal, etc)
- semânticas (os conceitos do domínio de aplicação que o termo pode expressar)

Existem variados formalismos para representação da informação que constitui os léxicos. Essa representação deve ser escolhida de acordo com a representação da gramática do sistema, uma vez que essas duas bases de conhecimento interagem durante o processamento do texto. O exemplo (1)

abaixo mostra duas entradas em um léxico representado em PATR-II [Shieber, 1984], um formalismo muito utilizado em sistemas para PLN.

Exemplo (1)

```
mesa
    <categoria> = substantivo
    <gênero> = feminio
    <número> = singular

comprou
    <cat> = verbo
    <tempo> = pretérito-perfeito
    <número> = singular
    <pessoa> = 3
    <arg1> = SN
    <arg2> = SN
```

Na segunda entrada do léxico, temos o verbo `comprar` na 3^a pessoa do singular do pretérito perfeito. As características (traços sintáticos) `<arg1> = SN` e `<arg2> = SN` indicam que este verbo tem como argumentos (sujeito e objeto direto) um `SN` (Sintagma Nominal – termo definido na seção 3.1 a seguir), e que portanto, é transitivo direto.

Alguns sistemas para PLN contam com apenas um léxico, que matém todos os termos utilizados pelo sistema. Em outros casos, encontramos um léxico geral, que é dado a priori, e um ou mais léxicos dedicados, montados pelo usuário com base no domínio específico da aplicação (*e.g.* domínio de computação, medicina, literatura, etc).

2.1.2 A Gramática

A gramática define, através de regras, quais são as cadeias de palavras válidas em uma língua. Essa verificação é feita em termos de categorias sintáticas, e não de uma lista exaustiva de frases – o que seria inviável, uma vez que qualquer língua possiu um número infito de frases gramaticalmente corretas.

Como no caso dos léxicos, existem aqui também diversos formalismos de representação, como veremos na seção 4.1. O exemplo (2) abaixo mostra uma regra gramatical no formalismo PATR-II. Esse tipo de regra traz restrições associadas, que funcionam como variáveis que terão seus valores determinados quando a regra for aplicada às palavras da frase.

Exemplo (2)

```
SN → Subst Adj
    <Subst gênero> = <Adj gênero>
    <Subst número> = <Adj número>
```

A regra acima determina que um Sintagma Nominal (`SN`) pode ser constituído de um substantivo seguido de um adjetivo (*e.g.*, `mesa branca`). As restrições `<Subst gênero> = <Adj gênero>` e `<Subs número> = <Adj número>` determinam que o gênero e o número do adjetivo e do substantivo devem concordar. Veremos maiores detalhes e exemplos sobre gramáticas na seção 4.1.

2.1.3 Outras Bases de Conhecimento

As outras bases de conhecimento de um sistema para PLN fornecem *o contexto* para o processamento de cada frase:

- O **Modelo do Domínio** fornece o contexto *enciclopédico*, armazenando conhecimento a respeito das entidades, relações, eventos, lugares, e datas do domínio, em algum formalismo de IA – *e.g.*, Lógica de Predicados, Redes Semânticas, Frames, Scripts ou Hierarquias de tipos [Winston, 1992].
- O **Modelo do Usuário** fornece o contexto *interpessoal*, armazenando conhecimento a respeito do usuário do sistema *e.g.*, os seus objetivos, planos, intenções, sua função, status, conhecimento do domínio, etc), através de representações como planejamento hierárquico ou atos da fala [Allen, 1995].
- O **Modelo do Discurso** fornece o contexto *textual*, armazenando informações sobre as frases previamente processadas para, por exemplo, auxiliar na interpretação ou geração de referências pronominais a entidades já introduzidas no discurso; esse modelo geralmente consiste em uma pilha contendo as caraterísticas sintáticas e semânticas das entidades já mencionadas (ver seção 4.3).

2.2 Comentários Finais

Apresentamos, nesta seção, a arquitetura do que seria o sistema para PLN ideal, o estado da arte atual. Devemos ressaltar, contudo, que os primieros sistemas para PLN que foram construídos dentro do paradigma de IA (dec. 70) eram constituídos por um único módulo de processamento, e utilizavam, em sua maioria, uma única base de conhecimento, que aglutinava informação sintática e semântica específica para o domínio da aplicação. Esses sistemas eram dedicados a um único domínio, criado apenas para testar o sistema. Em conseqüência, o esforço empreendido no desenvolvimento de cada sistema tinha que ser refeito cada vez que o sistema mudava de domínio.

Com a crescente necessidade de se produzirem sistemas comercialmente viáveis, houve uma tendência de se buscar portabilidade em sistemas computacionais, de maneira geral. No que concerne o PLN, sistemas modulares começaram a surgir com mais força na década de 80, tratando de domínios reais.

A seguir, serão apresentadas algumas definições lingüísticas que auxiliarão na compreensão deste texto.

3 Noções Básicas de Lingüística

Veremos nesta seção noções básicas de Lingüística indispensáveis à compreensão deste texto. Inicialmente, listamos os termos lingüísticos utilizados no texto, seguindo com uma breve apresentação de algumas áreas de estudo lingüístico.

3.1 Terminologia

Os termos listados a seguir foram definidos com base em [Dubois *et al.*, 1973] [Crystal, 1991] [Crystal, 1987] [Fromkin & Rodman, 1988].

Categoria sintática – termo que designa uma classe cujos membros pertecem ao mesmo "ambiente sintático". As palavras *cadeira*, *mesa* e *bolsa* podem figurar no mesmo ambiente sintático, por exemplo em: *Eu comprei uma* _. Exemplos de categorias sintáticas são: substantivo, adjetivo, verbo, preposição, determinante (definido abaixo), advérbio, etc.

Categorias fechadas – aquelas categorias que são constituídas por um número pequeno e fixo de palavras, por exemplo: artigos, preposições, conjunções, pronomes e interjeições.

Categorias abertas – substantivos, verbos, adjetivos e advérbios.

Sintagma – grupo de elementos lingüísticos classificado de acordo com a categoria sintática de seu elemento *núcleo*: os sintagmas nominais (SN) têm como elemento núcleo um (ou mais de um) substantivo; os sintagmas verbais (SV) têm um verbo ou uma locução verbal como núcleo; os preposicionais (SP) começam com uma preposição, os adverbiais (SAdv) têm um advérbio ou uma locução adverbial como núcleo, etc.

Os sintagmas são unidades lingüísticas de nível intermediário, sendo constituintes de uma unidade de nível superior, a frase. Exemplos:

- `SN` – João; o menino; a maçã verde; o gato de rabo longo.
- `SV` – chove; chegou cedo; tem estado doente; falaram de Maria a Pedro.
- `SP` – para você; de Maria a Pedro.
- `SAdv` – cedo; muito rapidamente.

Constituinte – toda palavra ou sintagma que constitui a frase. Exemplos:

- `O maestro viajou hoje.`
  `O maestro` → SN
  `O` → determinante
  `maestro` → substantivo;
  `viajou hoje` → SV;
  `viajou` → verbo;
  `hoje` → advérbio;

Determinantes – são constituintes do sintagma nominal, e dependem em gênero e número do substantivo que especificam (ou *determinam*). São eles: artigos, adjetivos, possessivos, demonstrativos, interrogativos, relativos e indefinidos, e numerais. Um determinante pode ser formado por mais de um constituinte, como em `'A bela moça'`, por exemplo.

Frase – unidade mínima de comunicação que tenha sentido completo. Pode conter um número qualquer de verbos. Exemplos:

- Fogo!
- Bom, este doce.
- O maestro viajou hoje.
- Quando cheguei, Maria já havia telefonado duas vezes.
- Está chovendo.

Oração – unidade sintática mínima que possui dois termos fundamentais: o *sujeito* e o *predicado*. Uma frase pode ser constituída de uma ou mais orações.

Papéis Sintáticos (ou **Funções Gramaticais)** – são as funções representadas pelas palavras ou sintagmas na estrutura gramatical das frases. As funções de *sujeito* e *predicado* definem as relações fundamentais da frase, havendo ainda os *complementos* – objetos diretos e indiretos, adjuntos nominais e adverbiais, etc. Exemplo:

- `Eu dei um livro a Pedro.`
  `Eu` → sujeito
  `Dei um livro a Pedro` → predicado
  `um livro` → objeto direto
  `a Pedro` → objeto indireto

Papéis Temáticos – papéis semânticos assumidos pelos constituintes da frase: *e.g.*, agente, instrumento, locação. Os papéis temáticos devem ser independentes dos papéis sintáticos dos componentes da frase (ver seção 4.2.2 para maiores detalhes). Assim sendo, as duas frases abaixo compartilham o mesmo agente, `Maria`, ainda que este seja sujeito em (1) e agente da passiva em (2):

(1) `Maria entregou-me o livro` (voz ativa)

(2) `O livro me foi entregue por Maria` (voz passiva)

3.2 Áreas de Estudo Lingüístico

As línguas naturais são formadas por um conjunto infinito de frases, que possuem uma realização sonora (a cadeia falada), e um significado (seu conteúdo semântico). As frases podem ser divididas em unidades menores de som e significado – as palavras, que por sua vez podem ser subdividas em unidades mínimas de som (os fonemas) e de significado (os morfemas).

A *gramática* de uma língua é um sistema formado por um componente fonético (os fonemas da língua e suas possíveis combinações), um componente morfológico (morfemas e suas regras de derivação), e um componente sintático, que constitui-se de regras que descrevem as estruturas base das frases sintaticamente corretas (gramaticais) naquela língua. A gramática é o "sistema" que cada falante de uma língua sabe e usa quando forma frases.

Veremos aqui um pouco de como a Lingüística trata do problema de representar e interpretar frases de uma língua, em isolamento ou em seqüência.

3.2.1 Fonética e Fonologia

A *Fonética* estuda os sons da fala, *i.e.*, as múltiplas realizações dos fonemas, provendo métodos para sua descrição, classificação e transcrição. A *Fonologia*, por sua vez, estuda o comportamento dos fonemas em uma determinada língua, ou seja, o sistema de som dessa língua.

No que concerne o PLN, a fonética e a fonologia dão suporte às pesquisas em *compreensão da fala*, que buscam desenvolver sistemas que possibilitem a comunicação com os computadores de forma oral.

3.2.2 Morfologia

Como dito acima, os morfemas são as unidades básicas de significado na composição das palavras. A *Morfologia* estuda a estrutura (ou *forma*) das palavras através dos morfemas, e suas leis de formação e inflexão.

Quanto ao seu significado, morfemas são classificados em *lexicais*, que têm significação própria (*e.g.*, sol, sabor, alegria), e *gramaticais*, cujo significado deriva das relações e categorias da língua (*e.g.*, 's' em ruas – marca de plural, 'a' em menina – marca de feminino).

É a partir dos morfemas lexicais, também chamados de radicais, que palavras da mesma família são geradas, e mantêm uma base de significado comum (*e.g.*, 'branc' em branco e branca). Quem dá

a variação entre essas palavras são os morfemas gramaticais, que podem ser uma desinência (gênero e número de substantivos, adjetivos e pronomes, número e pessoa dos verbos), um afixo (modificam o sentido do radical a que se agregam, *e.g.*, 'des' em desfazer), ou uma vogal temática (caracterizam a conjugação dos verbos – se 1^a (a), 2^a (e) ou 3^a (i,o)).

Estudos nesta área dão suporte à construção de verificadores ortográficos automáticos, por exemplo. Na verdade, a utilização de regras morfológicas possibilita que esses sistemas armazenem apenas os radicais das palavras, e derivem suas formas flexionadas pela aplicação das regras pertinentes.

3.2.3 Sintaxe

A *Sintaxe* estuda as regras que governam a formação de frases de uma determinada língua. Essas regras podem ser usadas para a determinação da estrutura sintática das frases geradas.

Uma frase é formada por constituintes (*e.g.*, SN, SV), que, por sua vez, são compostos por constituintes de ordem inferior (*e.g.*, SP, SAdv), até se chegar às categorias básicas (*e.g.*, substantivo, verbo, preposição, advérbio, etc).

Regras sintáticas determinam a ordem linear dos constituintes na frase, com base na sua categoria sintática. Os constituintes de uma frase mantêm uma relação de hierarquia entre si (*cf.* Figura 2). Veremos abaixo uma árvore que apresenta a estrutura sintática da frase `'A menina quebrou o jarro azul'` em termos de seus constituintes.

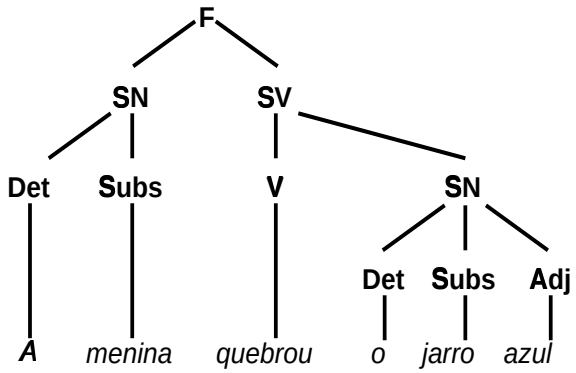

Figura 2: Estrutura sintática.

Nas *Gramáticas Gerativo-transformacionais*[3] [Chomsky, 1973], duas estruturas diferentes para cada frase são identificadas: a estrutura de superfície e a estrutura profunda.

Estrutura de superfície (ES) – é a organização sintática da frase tal como ela se apresenta, de modo que duas frases com o mesmo conteúdo e significado podem ter estruturas de superfície diferentes. Exemplo:

- `A menina quebrou o jarro azul.` (Fig. 2)
- `O jarro azul foi quebrado pela menina.`
  (F (SN (Det Subs Adj) SV (Vaux V SN (Det Subs))))

[3] criadas dentro do paradigam das Gramáticas Gerativas, descritas na seção 4.1.1.

Estrutura profunda (EP) – é a organização da frase em um nível mais semântico, referente à sua estrutura de base. Duas frases com ESs diferentes podem ter a mesma EP, como no exemplo acima. A EP pode ser representada em diferentes formalismos, como veremos na seção 4.2.

A determinação da estrutura sintática das frases é vista como uma etapa central na interpretação de LN, a partir da qual a frase de entrada pode ser formalmente analisada. Na geração de LN, ao contrário, o sistema é que gera a estrura sintática da frase de saída.

3.2.4 Semântica

A *Semântica* estuda o significado das palavras e como elas se combinam para formar o significado das frases. Isto é, enfoca a relação entre as unidades lingüísticas e o mundo.

Palavras têm propriedas semânticas que auxiliam na fixação do seu significado. Por exemplo, o traço sintático *feminino* é também uma propriedade semântica das palavras 'mesa', 'saudade' e 'menina'. A propriedade *humano* qualifica apenas 'menina' na lista acima. Podemos construir Ontologias de classes de palavras (seção 4.2.4) observando as características mais gerais das entidades existentes no mundo.

Não há, contudo, um sistema formal amplamente utilizado na representação semântica, como o que encontramos na sintaxe. Portanto, limitamo-nos aqui a apresentar alguns formalismos computacionais utilizados na interpretação semântica automática (seção 4.2).

3.2.5 Análise do Discurso

A *Análise do Discurso* estuda os princípios que governam a produção de seqüências estruturadas de frases (*discurso* – escrito ou falado). Fatores de *coesão* e *coerência* do discurso também são abordados aqui.

Esta área de estudo é especialmente importante na construção de sistemas de PLN por tentar indentificar qual a influência de uma ou mais frases na interpretação das frases subseqüentes. Este estudo é vital para interpretação de *pronomes* (*e.g.*, eu, você, ela, este, aquela), e *dêiticos* (*e.g.*, hoje, aqui, agora, etc).

A seção 4.3 apresenta um pouco do que é feito no processamento automático do discurso.

3.2.6 Pragmática

A *Pragmática* estuda a língua do ponto de vista de sua utilização na interação social – motivação pscicológica dos falantes, as restrições que eles encontram quando engajados em conversações, e o efeito que o uso da língua tem sobre os outros participantes em um ato de comunicação. Assim sendo, não é vista como mais um nível de representação estrutural da língua (como a representação sintática e a semântica).

Tanto a Pragmática quanto a Análise do Discursos têm como preocupação central a análise da conversação, e compartilham noções filosóficas e lingüísticas que abordam esse tópico. Como grande contribuição desta área, proveniente da Filosofia da Linguagem, temos a noção de *Atos da Fala* [Searle, 1971] (seção 4.3.1).

Os tópicos abordados aqui são muito variados, não havendo ainda uma teoria amplamente aceita nesta área. Alguns temas que constituem objeto de estudo são: estrutura do discurso, atos da fala, pressuposição, dêiticos, entre outros.

3.3 Comentários Finais

Os trabalhos nessas áreas de estudo lingüístico têm dado grande contribuição para o PLN, auxiliando no desenvolvimento de mecanismos que possibilitem a representação e o processamento de línguas por computador. O objetivo do PLN está mais voltado à construção de aplicativos do que à investigação do comportamento das línguas naturais – tarefa que fica a cargo da Lingüística Computacional (LC). Vale salientar, contudo, que os resultados obtidos pela LC são acompanhados de perto, e amplamente utilizados pelos pesquisadores em PLN.

Veremos a seguir três aspectos da Interpretação de Linguagem Natural: mecanismos para o processamento sintático de frases, formalismos para a interpretação semântica de LN, e processamento automático do discurso.

4 Interpretação de Linguagem Natural

Esta seção apresenta três aspectos da Interpretação de Linguagem Natural: (1) processamento sintático; (2) formalismos para interpretação semântica, e (3) processamento automático do discurso.

4.1 Processamento Sintático

O processamento sintático tem por objetivo obter uma representação da estrutura sintática da frase sob análise, que pode ser representada de variadas formas (*e.g.*, árvore sintática, constituintes da frase agrupados com colchetes, etc).

Para se determinar a estrutura sintática de uma frase, deve-se considerar essencialmente: (1) a *gramática* da língua, que especifica quais as estruturas válidas para essa língua, e (2) a técnica de *parsing* utilizada, *i.e.*, o método utilizado para determinar a estrutura sintática da frase sob análise. Um *parser* é um algoritmo que mapeia uma frase na sua estrutura sintática, utilizando para isso o léxico e a gramática do sistema.

Veremos abaixo uma breve descrição das gramáticas gerativas, seguida de alguns dos formalismos gramaticais e métodos de parsing mais utilizados em PLN. Esta seção foi escrita com base em [Allen, 1995] [Gazdar & Mellish, 1989] [Winograd, 1983].

4.1.1 Gramáticas Gerativas

Uma *gramática gerativa* (conceito derivado do estudo das linguagens formais) pode ser definida como um conjunto de regras que *geram* todas as cadeias válidas da linguagem por ela descrita, assinalando a cada cadeia sua estrutura sintática.

Uma gramática gerativa formal pode ser descrita por uma quádrupla $\mathtt{G} = <\mathtt{V}_t, \mathtt{V}_n, \mathtt{P}, \mathtt{F}>$, onde:

- $\mathtt{V}_t$ é o vocabulário terminal da linguagem (as palavras da língua);
- $\mathtt{V}_n$ é o vocabulário não-terminal da linguagem (as categorias sintáticas da língua);
- F é o símbolo inicial de G, onde F indica "Frase" e $\mathtt{F} \in \mathtt{V}_n$;
- P é um conjunto finito de regras de produção, na forma "$\alpha \rightarrow \beta$", onde α e β são cadeias de símbolos em $\mathtt{V}_t \cup \mathtt{V}_n$. P deve conter pelo menos uma regra $\mathtt{F} \rightarrow \alpha$.

O poder gerativo de uma gramática é determinado pela natureza de α e β em P, indicando que tipos de cadeias tal gramática é capaz de gerar. As gramáticas gerativas foram classificadas por Chomsky em quatro tipos [Chomsky, 1956].

- Tipo 3 – gramáticas regulares
- Tipo 2 – gramáticas livres-de-contexto
- Tipo 1 – gramáticas sensíveis ao contexto
- Tipo 0 – sistemas de reescrita geral

As gramáticas do tipo 3 são subconjuntos das gramáticas do tipo 2, que por sua vez são subconjuntos das gramáticas do tipo 1, que são subconjuntos das gramáticas do tipo 0. Gramáticas do tipo 3 são mais restritas, apresentando menor capacidade gerativa que as do tipo 2, 1 e 0, em ordem crescente. Em outras palavras, o poder gerativo de uma gramática é tão maior quanto menos restritivas forem suas regras.

Tipo	**Forma das produções**	**Linguagens**
3	α: um único símbolo não terminal β: um único símbolo terminal ou uma cadeia com um único terminal e um único não terminal.	Linguagens regulares
2	α: um único símbolo não-terminal β: uma cadeia de símbolos com um único terminal e/ou não-terminais.	Inclui linguagens tais como $X^nY^kZ^n$, mas não $X^nY^nZ^n$ ou WW, onde W é uma cadeia arbitrária de terminais e os dois W's são idênticos.
1	α e β: cadeias de símbolos terminais e não-terminais; a quantidade de símbolos em α deve ser menor ou igual à quantidade de símbolos em β; α deve conter pelo menos um símbolo não-terminal.	Linguagens sensíveis ao Contexto
0	α e β: cadeias de símbolos terminais e não-terminais.	Qualquer linguagem gerada por uma máquina computacional.

Tabela 1: Hierarquia de Tipos de Gramáticas.

Gramáticas do tipo 3, com poder gerativo fraco, não são capazes de definir linguagens simples com cadeias do tipo X^nY^n. Isto pode ser facilmente conseguido com gramática do tipo 2, com apenas duas regras *recursivas*: (1) $F \rightarrow XFY$ e (2) $F \rightarrow XY$.

O interesse dessas regras para PLN aparece quando lidamos, por exemplo, com orações relativas (que servem de complemento ao SN), como a seguir:

`O encanador que eu contratei chegou tarde`

onde $X =$ `O encanador` (SN), $F =$ `eu contratei` (oração relativa), $Y =$ `chegou tarde` (SV).

Gramáticas do tipo 3 são capazes apenas de reconhecer se determinada cadeia pertence ou não à linguagem descrita. Gramáticas do tipo 2, com poder gerativo forte, também associam uma estrutura a cada cadeia da linguagem.

Gramáticas do tipo 1 e 0 apresentam dificuldades na determinação da estrutura de determinadas cadeias, uma vez que admitem que α seja composto por mais de um símbolo, como nas cadeias abaixo:

- Tipo 1: AuB $\rightarrow$ ACB
- Tipo 0: ABCD $\rightarrow$ EF

4.1.2 Formalismos Gramaticais

A gramática de um sistema de PLN pode ser escrita em diversos formalismos, entre os quais citamos: de Transição [Woods, 1970], Gramáticas de Constituintes Imediatos (Phrase Structure Grammars - PSG), em particular, Generalised PSG [Gazdar *et al.*, 1985], Gramáticas de Casos (Case Grammars) [Fillmore, 1968], Gramáticas de Unificação Funcional (Functional-Unification grammars - FUG) [Kay, 1979] e PATR-II [Shieber, 1984]).

Veremos a seguir um pouco sobre os formalismos mais tradicionais da PLN: Gramáticas de Constituintes Imediatos e Redes de Transição.

4.1.3 Gramáticas de Constituintes Imediatos

Gramáticas de Constituintes Imediatos (Phrase Structure Grammars - PSG) são gramáticas livres-de-contexto cujos símbolos das regras de produção são categorias sintáticas (símbolos não terminais) ou palavras (símbolos terminais). A categoria da esquerda pode ser expandida (reescrita) pela(s) categoria(s) do lado direito da regra durante a derivação da estrutura da frase sob análise.

Estas gramáticas, portanto, provêem a estrutura sintática das frases em termos de seus constituintes, especificando a hierarquia entre eles. O exemplo (3), a seguir, ilustra uma pequena gramática capaz de gerar e reconhecer a frase `'A menina quebrou o jarro azul'`. A estrutura sintática dessa frase, representada em forma de árvore, aparece na figura 2.

Exemplo (3)

Gramática

```
F → SN SV
SN → Det Subs
SN → Det Subs Adj
SN → Subs Adj
SV → V SN

Det → a
Subs → menina
Subs → jarro
Adj → azul
V → quebrou
```

O grande interesse do PLN nas GLCs se deve ao fato de essas gramáticas, além de apresentarem forte poder gerativo, terem sido tradicionalmente utilizadas na ciência da computação com sucesso, havendo inúmeros algoritmos eficientes para reconhecer linguagens livres-de-contexto.

4.1.4 Redes de Transição

As Redes de Transição (RT), outro formalismos utilizado na representação de gramáticas, consistem em nós (representando estados) e arcos (representando categorias gramaticais). Essas redes tanto reconhecem se dada cadeia pertence a uma linguagem, como também provêem sua estrutura sintática (os arcos que foram atravessados durante o reconhecimento da cadeia).

Vejamos, por exemplo, a RT da figura 3 abaixo, com dois estados inicias `(1)` e `(2)`, e um estado final `(6)`.

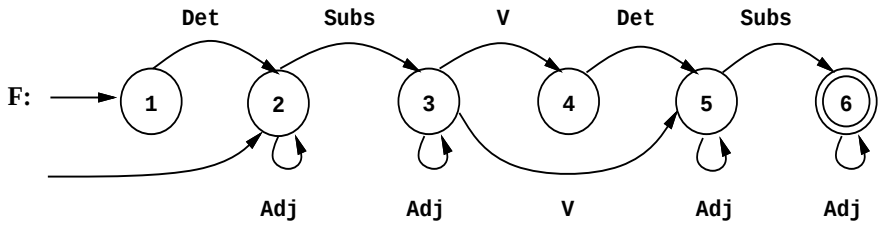

Figura 3: Rede de Transição Simples.

Esta rede reconhece frases como `'Maria chegou'`, ou `'O menino guloso comeu o bolo'`. Observem que *loops* (arcos que saem e chegam ao mesmo estado) representam categorias opcionais nas cadeias (*e.g.*, os adjetivos representados pelos arcos (2,2), (3,3), (5,5) e (6,6)).

O processo de reconhecimento se dá pelo *casamento* entre a categoria das palavras das cadeias e os rótulos dos arcos. A cadeia é dada como entrada, e as palavras são 'consumidas' uma a uma, na ordem em que aparecem na frase.

Por exemplo, a frase `'A menina quebrou o jarro azul'`, ao ser processada, atravessa os seguintes arcos:

`Det (1,2), Subs (2,3), V (3,4), Det (4,5), Subs (5,6), Adj (6,6).`

Contudo, a frase `'Está chovendo'` não será reconhecida pela gramática descrita por esta RT, uma vez que esta frase consiste em uma loucução verbal de verbo defectivo (não há sujeito), e a gramática descrita exige que qualquer cadeia válida comece sempre com um determinante, um substantivo ou um adjetivo.

Veremos a seguir três tipos de redes de transição, onde cada novo tipo é um refinamento do anterior no que concerne o seu poder gerativo.

(1) *Redes de Transição Simples* (RTS) – são autômatos de estado finito com estados iniciais e finais. Essas redes não aceitam arcos circulares que levem a um estado anterior ao ponto de partida do arco, podendo conter, no máximo, arcos circulares como os vistos na figura 3.

As RTS podem ser representadas por regras de produção como as usadas nas gramáticas PSG. A gramática equivalente à RTS da figura 3 é dada por:

`F` $\rightarrow$ `SN SV`
`SN` $\rightarrow$ `Det Subst`
`SN` $\rightarrow$ `Subst Adj`
`SV` $\rightarrow$ `V SN`

Limitações das RTSs: Essas redes têm a mesma capacidade gerativa das gramáticas regulares (tipo 3).

(2) *Redes de Transição Recursivas* (RTR) – são RTS cujos rótulos podem conter outra RT da gramática. RTRs têm a mesma capacidade gerativa das gramáticas livres-de-contexto (Fig. 4).

O efeito resultante de se percorrerem arcos que 'transferem' o processamento para uma sub-rede é obtido nas gramáticas livres-de-contexto através das chamadas recursivas.

Limitações das RTRs: Essas redes não são capazes de verificar concordância entre constituintes da frase, considerando como gramaticalmente corretas frases como: **'A menino quebraram uma jarro preta'**. Além disso, sofrem as mesmas limitações encontradas nas gramáticas-livres de contexto.

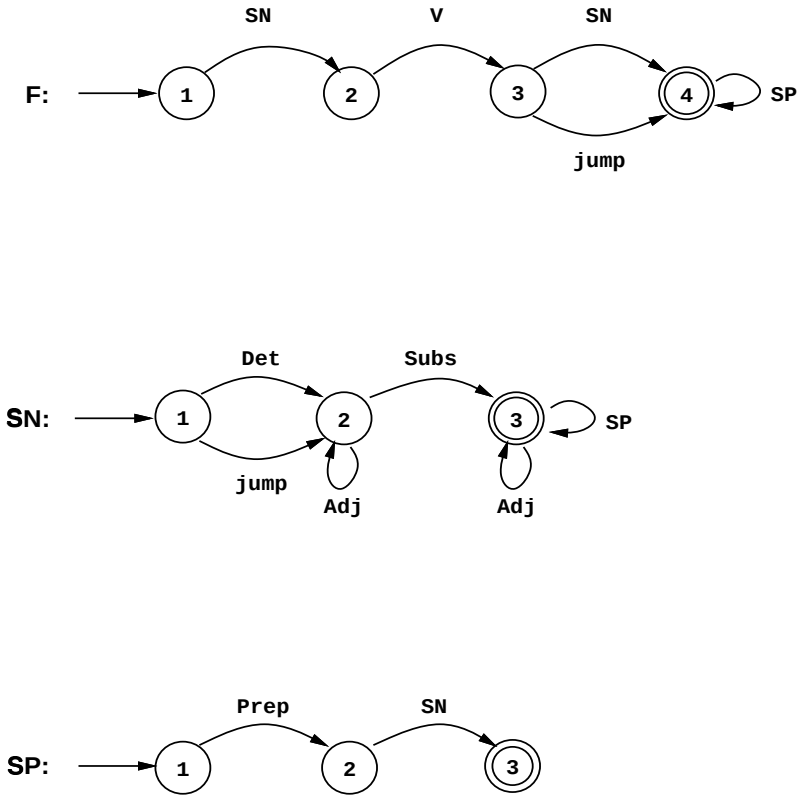

Figura 4: Rede de Transição Recursiva.

(3) *Redes de Transição Aumentadas* (RTA) – são RTRs acrescidas de *condições* e *ações* associadas aos arcos. A condição deve ser verdadeira para que o arco seja atravessado. A ação associada a um arco é responsável pela construção de uma estrutura descritiva do constituinte sob análise. Essas descrições parciais são armazenadas em registradores associados aos nós da árvore em construção. Registradores podem ser vistos como variáveis.

Se associarmos condições e ações às RTRs da figura 4, obteremos um conjunto de RTAs. A Tabela 2 abaixo ilustra as condições e ações associadas ao sintagma nominal da figura 4.

Arco	Condição	Ação
1-2	–	`Det` := determinante `Conc` := conc.det
2-3	Conc ∩ conc.det	`Núcleo` := substantivo `Conc` := conc.det ∩ conc.subs
1-3	–	`Subs` := substantivo `Conc` := conc.subs

Tabela 2: Condições e ações associadas a um sintagma nominal.

A seguir, temos a lista dos registradores utilizados pela RTA do exemplo acima:

`Det` – registra a existência de um determinante na cadeia sob análise.

`Conc` – guarda a concordância do sintagma: gênero, número e pessoa.

`Núcleo` – guarda a categoria do constituinte mais importante do sintagma, aquele que determina o tipo do sintagma.

`Subs` – registra a existência de um substantivo no sintagma.

A condição para se atravessar o arco `1-2` é que a cadeia se inicie com um determinante. A ação desencadeada é registrar-se a existência de um determinante, e armazenar-se sua concondância em outro registrador. No caso do arco `2-3`, a condição *default* do arco (*i.e.*, que o constituinte seja um substantivo) é *aumentada* pela condição extra de que a concordância de tal substantivo deva ser a mesma do determinante já encontrado. O arco `1-3` será atravessado se a cadeia se inicia com um substantivo, sem a ocorrência de um determinante.

As RTAs são, sem dúvida, o tipo de RT mais utlizado nos sistemas para PLN, por oferecerem, com suas *aumentações*, mais flexibilidade no tratamento das exceções encontradas nas LNs. Isto se deve ao fato de que as condições associadas aos arcos podem ser vistas como um *contexto* associado, o que faz essas redes equivalentes às gramáticas sensíveis ao contexto.

Críticas às RTAs: Essas redes são muito flexíveis, porém perdem em padronização. As condições e ações, em muitos casos, dependem da aplicação para a qual o sistema foi desenvolvido, comprometendo sua portabilidade para outra aplicação. Além disso, as RTAs são procedimentais – as ações são procedimentos que serão executados caso determinado arco seja atravessado – o que prejudica em muito a clareza descritiva dessas redes.

4.1.5 Gramáticas Computacionais

Ao contrário das de Transição, gramáticas são declarativas (ganhando em clareza descritiva), e são (tanto quanto possível) independentes do domínio da aplicação. Formalismos gramaticais para o PLN devem apresentar: (1) naturalidade lingüística; (2) forte poder gerativo; e (3) eficácia computacional.

Como vimos anteriormente, as gramáticas de constituintes imediatos livres-de-contexto oferecem forte poder gerativo e capacidade computacional. Contudo, precisamos de algo mais do que a determinação da estrutura sintática das frases. As cadeias do exemplo (4) abaixo, por exemplo, serão todas consideradas sintaticamente corretas por gramáticas que não considerem os *traços sintáticos* dos constituintes:

Exemplo (4)

```
O menino anda
Os meninos andam
* O menino falam
* Os menino anda
```

PATRII

Uma das maneiras de se verificar a concordância de gênero e número entre os constituintes da frase é obtida através do formalismo PATRII, já mencionado anteriormente. Neste caso, as regras da gramática trazem traços sintáticos associados, espécie de variáveis a serem instanciadas de acordo com as características das palavras da frase. No exemplo abaixo, apresentamos o mesmo léxico e a mesma gramática do exemplo (3), agora no formalismo PATRII:

Exemplo (5)

Gramática

```
F → SN SV
     <SN número> = <SV número>
     <SN pessoa> = <SV pessoa>
     <SN caso> = nominativo
     <SV forma> = flexionado
SN → Det Subs
     <Det gênero> = <Subs gênero>
     <Det número> = <Subs número>
SN → Det Subs Adj
     <Det gênero> = <Subs gênero>
     <Det número> = <Subs número>
     <Subs gênero> = <Adj gênero>
     <Subs número> = <Adj número>
SV → V SN (verbo transitivo direto)
     <SN caso> = objeto direto
```

Léxico

```
a
     <categoria> = Determinante
     <gênero> = fem
     <número> = sing
menina
     <categoria> = Substantivo
     <gênero> = fem
     <número> = sing
jarro
     <categoria> = Substantivo
     <gênero> = fem
     <número> = sing
azul
     <categoria> = Adjetivo
     <número> = sing
quebrou
     <cat> = Verbo
     <tempo> = pass
     <número> = sing
     <pessoa> = 3
     <arg1> = SN
```

Durante o *parsing* da frase, os valores dos traços sintáticos das palavras (fornecidos pelo léxico) são utilizados para fixar os valores das variáveis associadas às regras da gramática. Assim, trona-se possível a verificação da corretude gramatical da frase.

4.1.6 Métodos de Parsing

Podemos dizer, com base no que foi visto até agora, que o objetivo central de uma gramática (na interpretação de LN) é atribuir estruturas sintáticas a frases que constituam cadeias válidas

para a linguagem descrita por essa gramática. Veremos nesta seção como obter essas estruturas automaticamente.

Como dito anteriormente, um *parser* é um algoritmo que mapeia uma frase na sua estrutura sintática, utilizando para isto o léxico e a gramática do sistema. O léxico informa a(s) categoria(s) sintática(s) de cada palavra que compõem o domínio do sistema, enquanto a gramática determina quais as cadeias de categorias válidas para a linguagem por ela descrita. A construção dos algoritmos de *parsing* está, portanto, intimamente ligada ao formalismo gramatical utilizado.

Antes de apresentarmos as estratégias de parsing, seria interessante discutirmos um pouco sobre um problema que dificulta imensamente a análise automática das línguas naturais: a ambigüidade.

Ambigüidade

No nível sintático, podemos identificar duas formas de ambigüidade: léxica e estrutural.

A ambigüidade léxica se dá quando uma palavra na frase pode assumir mais de um significado (exemplo (6)). A ambigüidade estrutural, por outro lado, se dá quando a mesma frase pode ser mapeada em mais de uma estrutura sintática válida para a mesma interpretação (exemplo (7)). Esse tipo de ambigüidade só pode ser tratada por gramáticas com capacidade gerativa forte, que sejam capazes de gerar mais de uma estrutura sintática para a mesma cadeia de entrada.

Exemplo (6): Ambigüidade Léxica

`Ela estava em minha companhia.`

1. `companhia = empresa`
2. `companhia = pessoa` (*i.e.*, ela estava comigo)

Exemplo (7): Ambigüidade Estrutural

`Eu vi o rapaz no parque com o binóculo`

`(1) O rapaz estava com o binóculo`

`(2) Eu estava com o binóculo`

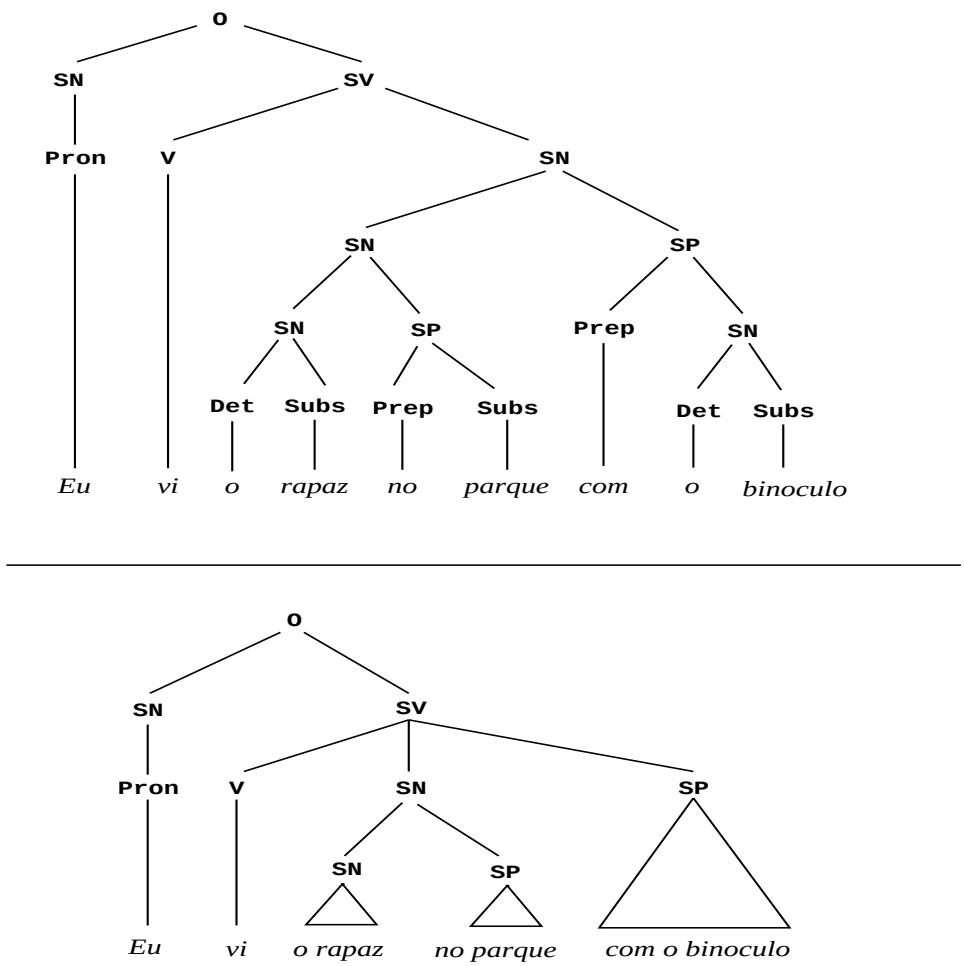

Figura 5: Exemplo de ambigüidade estrutural.

Veremos a seguir duas estratégias de parsing – top-down e bottom-up, e como os algoritmos lidam com o problema da ambigüidade.

4.1.7 Parsing Top-Down

Como dito acima, o *parser* é um algoritmo que mapeia a frase de entrada em uma estruruta sintática. Em casos de ambigüidade, podemos obter mais de uma estrutura como resultado. As categorias sintáticas das palavras estão disponíveis no léxico, e a gramática descreve as combinações válidas para uma dada linguagem.

Utilizaremos uma gramática de constituintes imediatos para auxiliar na apresentação dos métodos de parsing. Ressaltamos, contudo, que as redes de transição também podem ser utilizadas para o mesmo fim.

Considere a gramática e o léxico no exemplo abaixo para se fazer o parsing da frase `A menina quebrou o jarro azul`:

Exemplo (8)

Gramática

1. `F` → `SN SV`
2. `SN` → `Det Subst`
3. `SN` → `Det Subs Adj`
4. `SV` → `V SN`

Léxico

`Det` → `a`
`Subs` → `menina`
`Subs` → `jarro`
`Adj` → `azul`
`V` → `quebrou`

Um parser *top-down* começa o processamento com o símbolo `F` e tenta reescrever este símbolo com base nas regras disponíveis na gramática. No caso acima, `F` pode consistir em um `SN` seguido de um `SV`.

Em seguida, o parser procura uma regra que substitua o `SN` encontrado, e então teremos duas possibilidades:

1. a segunda regra reescreve o `SN` como `Det` seguido de `Subs`
2. a terceira regra reescreve o `SN` como `Det` seguido de `Subs` seguido de `Adj`

Contudo, a segunda possibilidade é descartada porque a frase sob análise se inicia com um determinante seguido de um substantivo, como apresentado pela opção (a) acima.

O `SV`, nesta gramática, só pode ser reescrito como um verbo seguido de um `SN`. O `SN`, como visto, apresenta duas possibilidades de reescrita. Nesta etapa, contudo, a opção de reescrita escolhida é dada pela regra (3) da gramática acima.

A partir daí, devemos buscar palavras no léxico que sejam da categoria sintática de cada palavra da cadeia, até que a cadeia só apresente itens lexicais. A figura 6 ilustra o processo descrito aqui, isto é, a árvore de busca do parsing da frase `'A menina quebrou o jarro azul'`, representada na figura pelas letras iniciais de cada palavra na frase.

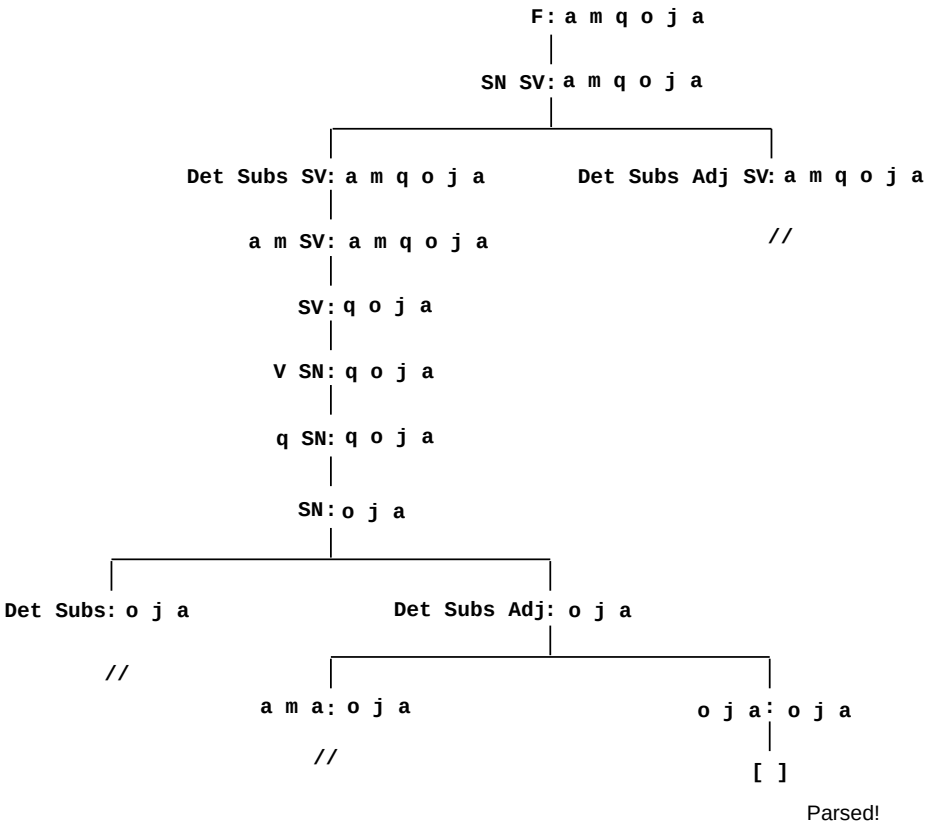

Figura 6: Árvore de busca do parsing top-down de `A menina quebrou o jarro azul`.

4.1.8 Parsing Bottom-Up

Este método de parsing trabalha na direção inversa à do parser *top-down*. Aqui, o algoritmo parte das palavras e tenta chegar a uma estrutura sintática para a frase. Consideremos a frase `Maria casa hoje`, com a gramática e o léxico do exemplo abaixo:

Exemplo (9)

Gramática

1. F → SN SV
2. SN → Subs
3. SN → Det Subs
4. SV → V
5. SV → V Adv

Léxico

```
Subs → Maria
Subs → casa
V → casa
Adv → hoje
```

O algoritmo inicia seu processamento verificando no léxico a categoria sintática da primeira palavra da frase (no caso, `Maria - Subs`). Em seguida, o algoritmo busca, entre as regras da gramática, alguma cujo lado direito 'case' com esta categoria. Podemos então reescrever `Subs` como `SN` (Fig. 7). Observe que o objetivo aqui é partir das palavras da frase e chegar até o símbolo `F`.

O algoritmo prossegue tentando reescrever o símbolo `SN`, mas não encontra nenhuma regra onde ele apareça sozinho do lado direito. Neste caso, o algoritmo busca a categoria da próxima palavra na frase, no caso, `casa`. Aqui, deparamo-nos com um caso de ambigüidade léxica, uma vez que o dicionário apresenta duas entradas (com categorias sintáticas diferentes) para esta palavra. Portanto, teremos mais de uma possibilidade a analisar (Fig. 7).

1. `'casa'` pode ser um `Subs`
2. `'casa'` pode ser um `V` flexionado

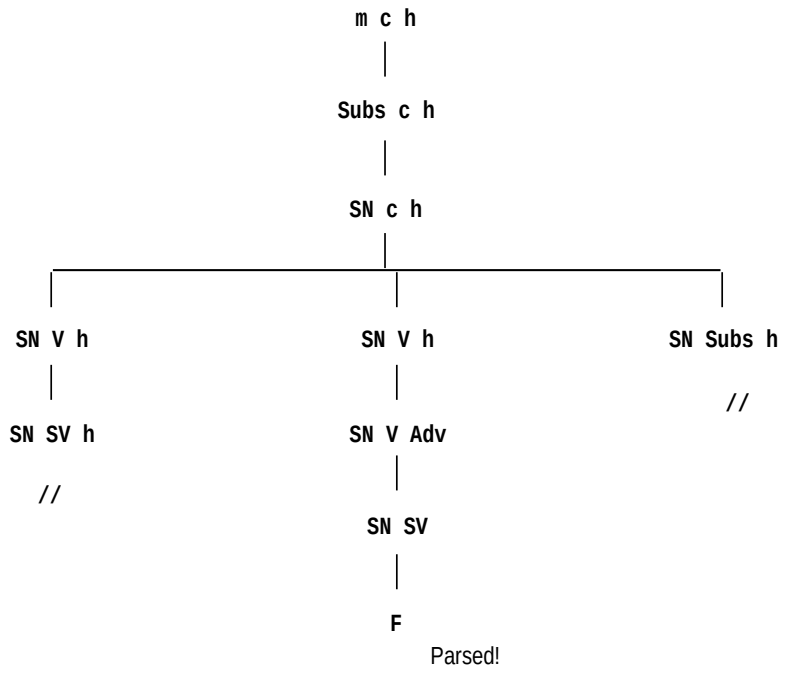

Figura 7: Árvore de busca do parser bottom-up para `'Maria casa hoje'`.

Seguindo a primeira opção, o algoritmo reescreve `Subs` como `SN`. Como não há mais nenhuma regra cujo lado direito consista apenas em `SN`, o algoritmo então passa a analisar a possibilidade de combinar os dois símbolos de que dispomos no momento: `SN` seguido de `SN`. Como não existe nenhuma regra que combine essas duas categorias, o algoritmo passa à próxima palavra, `hoje`.

Teremos então a seqüência (`SN, SN, Adv`), que não pode ser reescrita como `F`. O algoritmo, então, abandona esta opção e passa a analisar a opção onde `casa` foi reescrito como verbo.

Analisando agora a segunda opção, verificamos que `V` pode ser reescrito como `SV` que, combinado com o `SN` já obtido, pode ser reescrito como `F`. O processo de parsing teria terminado com sucesso se não fosse pelo `Adv` que aparece no final da frase. Mais uma vez, o algoritmo abandona o caminho sendo seguido e tenta outra combinação de categorias.

Finalmente, o algoritmo combina `V` com `Adv`, chegando a `SV`. Combinado então o `SN` existente com o `SV` obtido, chegamos a `F` sem que reste nehuma palavra da frase a ser processada. O algoritmo termina com sucesso, devolvendo a estrutura sintática da frase de entrada (Fig. 7).

4.1.9 Comentários Finais

O problema da determinação da estrutura sintática de frases não é tão simples como os exemplos acima podem sugerir. As gramáticas utilizadas nos exemplos (8) e (9) acima são muito simples, contando com poucas regras. Isto restringe imensamente as possibilidades de se gerarem estruturas sintáticas variadas.

Por outro lado, quando se aumenta a quantidade de regras da gramática (a fim de aumentar sua abrangência), tem-se, como efeito colateral, um aumento substancial de opções a seguir durante o parsing da frase. Caso não exista um controle efetivo, o parser poderá gerar muitos caminhos que não levam a uma estrutura sintática da frase, o que aumenta o tempo de processamento, diminuindo a eficiência do parser.

4.2 Interpretação Semântica

A obtenção da estrutura sintática das frases é, sem dúvida, uma fase crucial em qualquer sistema de interpretação de LN. Contudo, essa estrutura não é suficiente no processo de construção de um sistema para interpretação de linguagem natural. Podemos observar que as duas frases abaixo possuem palavras e estruturas de superfície diferentes, porém carregam o mesmo significado:

Exemplo (10)

```
(1) Eu dei um livro a Maria.
    ▷ estrutura sintática:  [Pron V Det Subs Prep Subs]
(2) Maria recebeu um livro de mim.
    ▷ estrutura sintática:  [Subs V Det Subs Prep Pron]
```

Um sistema de PLN deveria ser capaz de, a partir da frase (1), inferir o conteúdo da frase (2), para responder a uma pergunta como: `O que Maria recebeu de mim?`. Esta tarefa, contudo, é muito complexa, e envolve o significado das palavras, incluindo o conhecimento de que o ato de 'receber' e o ato de 'dar' são duas maneiras de se expressar o mesmo conceito de *transferência* de um objeto entre dois 'possuidores'.

Veremos nesta seção um pouco sobre representação semântica de palavras, e como o significado de cada palavra influi na interpretação de outras palavras na frase. Iniciamos com as Gramáticas de Casos, seguindo com restrições de seleção e hierarquias de tipos. Temos a seguir as redes semânticas e a representação semântica baseada em lógica de predicados.

4.2.1 Formalismos para Representação Semântica

Antes de iniciarmos nossa apresentação, seria importante fazermos uma 'classificação' dos formalismos aqui descritos no que concerne seu grau de representação.

Classificamos formalismos para representação do conhecimento, de modo geral, entre *fracos* e *fortes* [Rich & Knight, 1994]:

1. *fracos:* aqueles formalismos que fornecem apenas a *forma* de representar o conhecimento (*e.g.*, redes semânticas, frames).
2. *fortes:* aqueles que, além da forma, trazem também parte do conhecimento a representar pré-definido (*e.g.*, gramáticas de casos, scripts, dependência conceitual).

No que concerne formalismos para representação semântica, podemos ainda fazer distinção entre representações *superficiais* ou *profundas*.

1. *superficiais:* aquelas baseadas em generalizações lingüísticas pertinentes a várias línguas, e que sejam independentes do domínio de aplicação (*e.g.*, gramáticas de casos, dependência conceitual).
2. *profundas:* aquelas baseadas em formalismos da Inteligência Artificial para representar conhecimento especializado sobre um domínio específico (*e.g.*, scripts, frames).

A interpretação de uma frase poderá ser construída em duas etapas: (1) mapeando-se sua estrutura sintática (árvore de parsing) em uma representação semântica superficial, e em seguida (2) mapeando-se essa representação superficial em uma representação semântica profunda. A primeira etapa deste processo pode ser independente do domínio, e portanto portável; a segunda etapa, contudo, é necessariamente dependente do domínio.

Apresentamos a seguir alguns dos formalismos citados acima.

4.2.2 Gramáticas de Casos

As Gramáticas de Casos são muito utilizadas no processamento semântico de frases. Estas gramáticas, diferentemente das gramáticas sintáticas, não utilizam a noção de sujeito, objeto, etc. Aqui, *papéis temáticos*, ou *casos*, são atribuídos aos constituintes da frase.

Vemos a seguir alguns papéis temáticos apresentados por [Allen, 1995].[4]

Agente – o ser animado que causa a ação

Co-agente – agente secundário na ação

Tema (ou Paciente) – a coisa afetada peloa ação ou sobre cuja existência se discute

Instrumento – força ou instrumento usado para causar a ação

Beneficiário – a pessoa para quem a ação é realizada

Locação – o lugar onde a ação ocorre

Destinação – locação final

Fonte – locação de origem

Possuidor – possuidor do tema

Recipiente – possuidor final

[4] Existem classificações diversas, a depender do autor que as propõe.

Tempo – tempo em que a ação ocorre

O verbo, o constituinte central da frase, determina os papéis temáticos que servem de argumento para ele. Os verbos `comprar` e `botar`, como em `'Eu comprei um livro'` e `'Eu botei o livro sobre a mesa'`, teriam as seguintes entradas lexicais:

Exemplo (11)

- `comprar, Verbo -- SN` *(argumentos: agente, tema)*
- `botar, Verbo -- SN, SP` *(argumentos: agente, tema, locação)*

O resultado desta classificação pode ser visto nas tabelas a seguir:

`Papel temático`	agente		tema
`Papel sintático`	sujeito	núcleo do predicado	obj. indireto
`Categoria sintática`	pronome	verbo	SP
`Frase 1`	*Eu*	*comprei*	*um livro*

`Papel temático`	agente		tema	locação
`Papel sintático`	sujeito	núcleo do pred.	obj. direto	adjunto adv. lugar
`Categoria sintática`	pronome	verbo	SN	SP
`Frase 2`	*Eu*	*botei*	*o livro*	*sobre a mesa*

Os demais papéis da frase são determinados com base em estruturas mais completas, que prevêem a existência de 'complementos' para os verbos, além dos objetos que aparecem nas entradas mostradas acima. A frase `'Ontem eu botei o livro sobre a mesa apressadamente'` apresenta dois advérbios: um de tempo e outro de modo. Seus papéis temáticos são apresentados abaixo:

`Papel tem.`	tempo	agente		tema	locação	modo
`Papel sint.`	adj. adverbial	sujeito	núcleo	obj. dir.	adj. adv.	adj. adv.
	de tempo		pred.		de lugar	de modo
`Cat. sint.`	advérbio	pron.	verbo	SN	SP	advérbio
`Frase`	*Ontem*	*eu*	*botei*	*o livro*	*sobre a mesa*	*apressadamente*

Os papéis temáticos devem ser independentes da estrutura de superfície da frase, por serem vistos como *relações* entre constituintes da estrutura profunda da frase. Portanto, frases com significados diferentes devem ter representações distintas, e frases com o mesmo significado devem resultar na mesma representação.

Exemplo (12)

1. `Maria entregou o livro novo a João.`
2. `Maria entregou a João o livro novo.`
3. `O livro novo foi entregue a João por Maria.`

Nas três frases acima, cujas estruturas de superfície diferem na disposição de um ou mais constituintes, os papéis temáticos permanecem invariantes. Nos três casos, `Maria` é *agente*, `livro` é o *tema*, e `João` é *beneficiário*.

`Papel tem.`	agente		tema	beneficiário
`Papel sint.`	sujeito	núcleo pred.	obj dir	obj ind
`Cat. sint.`	nome p.	verbo	SN	SP
`Frase 1`	*Maria*	*entregou*	*o livro novo*	*a João*

Papel tem.	agente		beneficiário	tema
Papel sint.	sujeito	núcleo pred.	obj ind	obj dir
Cat. sint.	nome p.	verbo	SP	SN
Frase 2	*Maria*	*entregou*	*a João*	*o livro novo*

Papel tem.	tema		beneficiário	agente
Papel sint.	sujeito	núcleo pred.	obj ind	agente da passiva
Cat. sint.	SN	verbo	SP	SP
Frase 3	*O livro novo*	*foi entregue*	*a João*	*por Maria*

Vale lembrar que a lista de papéis temáticos aqui apresentada é uma opção entre muitas outras encontradas na literatura. Variações são possíveis tanto nos papéis propostos, quanto na forma de classificar os componentes em um papel ou outro. Outras classificações são encontradas em [Halliday, 1994] [Fawcett, 1987] [Jackendoff, 1990] [Talmy, 1985] [Quirk *et al.*, 1985].

4.2.3 Restrições de Seleção

Restrições de seleção são atribuídas às palavras no léxico de um sistema de PLN, a fim de restringir as possibilidades de combinações entre palavras. Essas restrições constituem *traços semânticos* que podem ser associados a entradas lexicais, nos mesmos moldes dos traços sintáticos e morfológicos.

Na interpretação de LN, essas retrições auxiliam na eliminação de ambigüidade léxica. A palavra `'cabo'`, por exemplo, pode significar uma patente militar ou um fio elétrico. Com base nas retrições de seleção, o sistema deveria ser capaz de, ao encontrar uma frase como `'O cabo é jovem'`, escolher o primeiro significado para a palavra `'cabo'` em detrimento do segundo.

Na geração de LN, essas restrições auxiliam na escolha das palavras a serem utilizadas para realizar o conteúdo a ser expresso.

O exemplo (13) abaixo ilustra entradas lexicais com restrições de seleção associadas. Essas restrições são associadas aos substantivos e a palavras de outras classes sintáticas que se combinam com os substantivos, a fim de filtrar as combinações possíveis.

Exemplo (13)

- `mulher` → `[+humano], [+feminino], [+adulto]`
- `menino` → `[+humano], [+masculino], [-adulto]`
- `pensamento` → `[+abstrato]`
- `cabo` → `[+vivente], [+humano], [+adulto]`
- `cabo` → `[+concreto], [+inanimado]`
- `jovem` → `Substantivo`[5] `[+vivente]`
- `morrer` → `Substantivo`[6] `[+vivente]`

Os adjetivos trazem associadas as restrições que devem aparecer nos substantivos por eles modificados, como é o caso de jovem (acima), que qualifica substantivos 'viventes'. No caso dos verbos, as retrições estão relacionadas aos seus argumentos: 'morrer', por exemplo, só pode ser usado com sujeitos 'viventes'.

O léxico pode também conter regras de redundância e postulados semânticos, como por exemplo:

[5] modificado pelo adjetivo.
[6] sujeito do verbo.

- Regras de redundância
  - [+humano] → [+animado]
  - [+humano] → [-abstrato]
  - [+lento] → [-rápido]
- Postulados semânticos (Lógica de predicados)
  - metal(x) → concreto(x)
  - possui(x,y) → pertence-a(y,x)

Os postulados semânticos podem ser usados para se deduzir, por exemplo, que dar(x,y,z) → receber(z,y,x), o que nos auxiliaria a interpretar o exemplo (10) acima.

4.2.4 Hierarquias de Tipos

Restrições de seleção definem classes semânticas de palavras, que podem ser organizadas hierarquicamente (Fig. 8).

Hierarquias de tipos, um tipo de ontologia, são uma maneira de representar conhecimento sobre a estrutura do mundo. Os tipos de cada objeto informam suas características. Do exemplo da ontologia acima (Fig. 8), podemos concluir que um cachorro é um objeto físico vivente e animal.

Algumas palavras pertencem a mais de uma classe ao mesmo tempo, por terem mais de um significado (*e.g.*, planta – vegetal vegetal ou projeto arquitetônico - Documento).

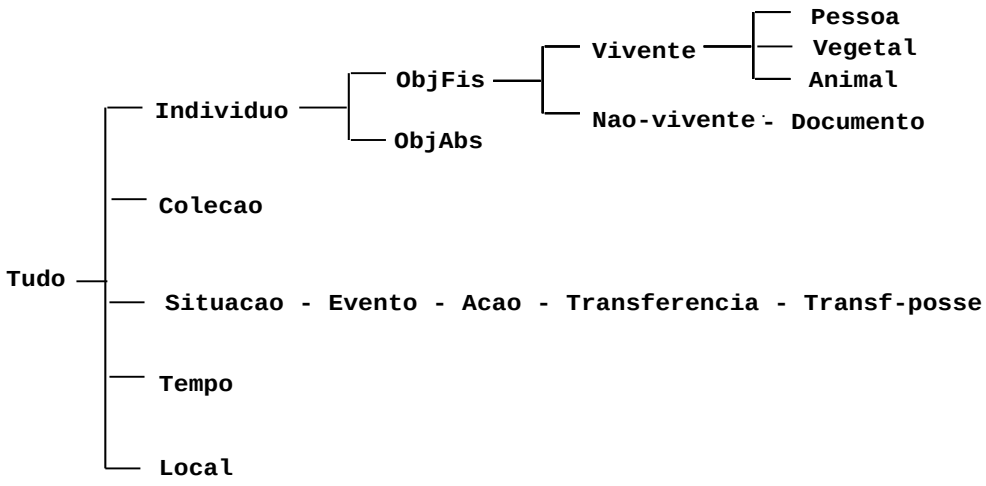

Figura 8: Hierarquia de tipos para classificar palavras.

4.2.5 Redes Semânticas

Redes semânticas são outro formalismo que também pode ser usado na representação semântica de frases. Uma rede semântica é uma estrutura de nós e arcos onde os *nós* representam entidades do domínio e os *arcos* representam relações entre essas entidades. Vejamos o exemplo da figura 9 abaixo:

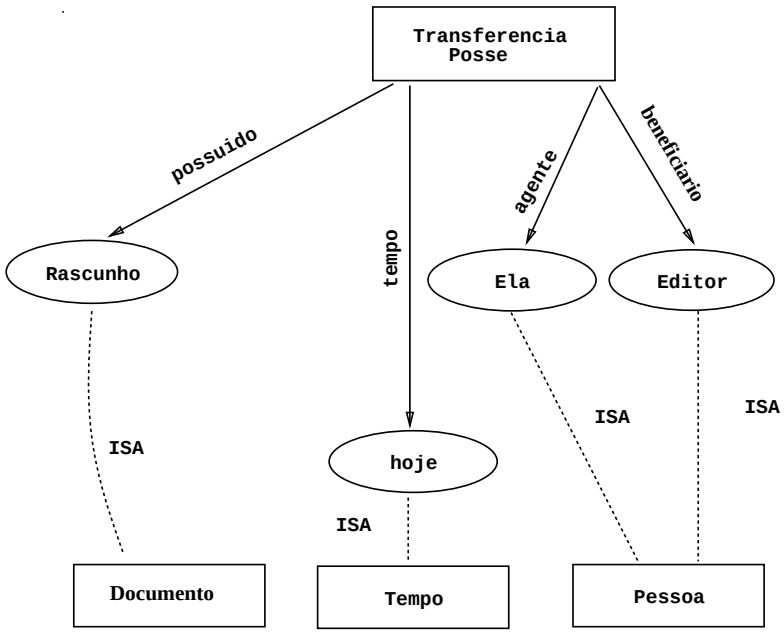

Figura 9: Rede semântica usada na representação de LN.

Uma das maiores vantagens oferecidas por essas redes é a sua capacidade de suportar *herança* de características entre entidades relacionadas por um tipo especial de relação – a relação *is-a* (é-um). Esta relação indica que uma entidade pertence a uma classe mais alta na hierarquia, ou pertence a uma categoria de objetos e, por conseguinte, *herda* as características dessa classe ou categoria – como na programação orientada a bojetos (*cf.* Fig. 9).

Redes semânticas podem ser utilizadas para representar ontologias, como a vista na figura 10, onde os nós guardam os tipos e os arcos representam a relação de subtipo.

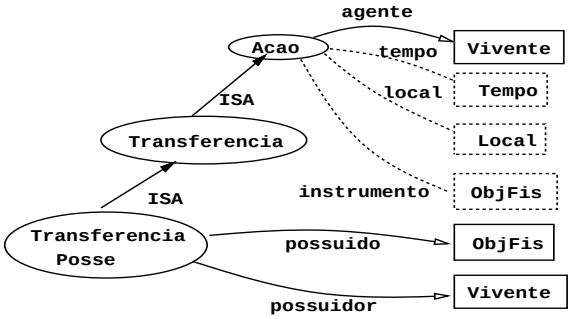

Figura 10: Rede Semântica equivalente ao subtipo `Ação`.

Ressaltamos que o `possuidor` na rede do exemplo acima não é necessariamente um ser `vivente`; dizemos, por exemplo, que uma empresa 'possui' patrimônio, empregados, etc. A Fig 10 traz uma

rede simplificada.

4.2.6 Fórmulas Lógicas

A Lógica de Predicados também tem sido amplamente utilizada na representação semântica de LN. As palavras do léxico são as *constantes* desta representação. Constantes que descrevem objetos (concretos ou abstratos – como eventos ou situações) são chamados de *termos*. Constantes que descrevem relações e propriedades são chamadas de *predicados*. Uma *proposição* é formada por um predicado seguido de um número de termos, seus argumentos. Vejamos o exemplo (14) abaixo:

Exemplo (14)

- Frase: `Rex é um cachorro.`
  Fórmula lógica: `cachorro(rex)`
- Frase: `Maria caiu.`
  Fórmula lógica: `caiu(maria)`
- Frase: `Ela entregou o livro ao editor.`
  Fórmula lógica: `entregar(ela, livro, editor)`

Fazendo-se um paralelo entre a representação em fórmulas lógicas e as redes semânticas, observa-se que os nós dessas redes (que representam entidades do domínio) são representados aqui por termos, enquanto que os arcos das redes são equivalentes aos predicados nas fórmulas.

Predicados que só apresentam um argumento são chamados de predicados unários (ou propriedades) – *e.g.*, `homem(pedro)`, enquanto predicados com n argumentos são chamados de predicados n-ários – *e.g.*, `comprar(maria,livro)`.

Podemos observar que palavras que pertencem a classes diferentes correspondem a tipos diferentes de constantes. Nomes próprios são termos, enquanto que nomes comuns (*e.g.*, `cachorro`) podem aparecer como termos ou como predicados unários, a depender da frase. No exemplo acima, `cachorro` aparece como predicado unário. Verbos, por sua vez, são predicados n-ários, onde n é dado pelo número de argumentos que o verbo pode ter. No caso da terceira frase do exemplo acima, a forma verbal `entregou` é representada por um predicado 3-ário, uma vez que apresenta um sujeito e dois objetos – `entregou(ela,livro,editor)`.

O exemplo acima mostra frases muito simples, que podem ser representadas por um predicado e termos. Para proposições mais complexas, utilizam-se os *operadores lógicos*:

$\neg$ (negação)
$\wedge$ (conjunção)
$\vee$ (disjunção)
$\Rightarrow$ (implicação)

Vejamos exemplos de uso desses operadores no exemplo (15) a seguir:

Exemplo (15)

- Frase: `André comeu e bebeu.`
  Fórmula lógica: `comeu(andre)` $\wedge$ `bebeu(andre)`
- Frase: `Carlos não ama Teresa.`
  Fórmula lógica: $\neg$ `(ama(carlos,teresa))`

Esses operados permitem que um maior número de frases seja representado. Contudo, ainda não são suficientes para representar frases como, por exemplo, `Todo homem é mortal`. Para estes casos, utilizamos outros construtores semânticos, os *quantificadores*. Existem apenas dois quantificadores no cálculo de predicados de primeira ordem, no qual se baseia esta apresentação: o universal ($\forall$), e o existencial ($\exists$).

Quantificadores modificam *variáveis*, termos que assumem valores dinamicamente, e que estão limitados pelo *escopo* (alcance) do quantificador que os modifica. Podemos, então, escrever fórmulas como:

(1) $\exists(x).P(x)$, que é verdadeira *sse* $P(x)$ é verdade para algum elemento do domínio;

(2) $\forall(x).P(x)$, que é verdade *sse* $P(x)$ é verdade para todos os elementos no domínio.

Exemplo (16)

- Frase: `Todo homem é mortal.`
  Fórmula lógica: $\forall$`(x).(homem(x)` $\rightarrow$ `mortal(x))`
- Frase: `Um homem morreu.`
  Fórmula lógica: $\exists$`(x).(homem(x)` $\wedge$ `morreu(x))`

Na primeira frase, o determinante `todo` é representado pelo quantificador $\forall$, cujo escopo alcança as duas ocorrências da variável x, e pelo operador $\rightarrow$. Na segunda frase, o determinate `um` é repesentado pelo quantificador $\exists$ e pelo operador $\wedge$.

Os quantificadores $\forall$ e $\exists$ são insuficientes para representar todos os quantificadores de uma língua, como por exemplo, *a maioria, alguns, muitos, poucos*, etc. Esse problema pode ser tratado por construtores chamados de *quantificadores generalizados* (exemplo (17)).

Exemplo (17)

- Frase: `A maioria dos alunos passou no curso.`
  Fórmula lógica: `MAIORIA(x).(aluno(x)` $\wedge$ `passou-no-curso(x))`

O problema aqui é "como interpretar este quantificador". Para este exemplo simples, pode-se resolver o problema contando os elementos do domínio que são alunos aprovados no curso, e comparando esse total com o total de alunos reprovados. Contudo, o mesmo não se aplica ao caso do quantificador *muitos*, por exemplo.

A fim de utilizarmos as fórmulas lógicas como uma representação semântica para frases, é necessário estabelecer-se uma correspondência entre as constantes e variáveis dessas fórmulas e os conceitos do domínio que elas representam. Contudo, este mapeamento não é trivial. Vejamos o caso da frase `'A menina loura é bonita'`. Para um domínio que apresenta mais de uma menina loura bonita, qual delas será escolhida como referente para a constante `menina`?

Finalmente, observamos que as fórmulas lógicas nos dizem se uma dada frase é `verdadeira` ou `falsa` em relação ao domínio tratado. Definição de classes de objetos e hereditariedade não são claros nessa representação.

4.2.7 Comentários Finais

Existe ainda um grande número de formalismos utilizados para representação semântica, entre os quais destacamos [Winston, 1992] [Rich & Knight, 1994]:

- Frames, que são especializações de Redes Semânticas;
- Scripts, estruturas semelhantes aos frames, utilizados na representação de seqüências de eventos estereotipadas (*e.g.*, ir a um restaurante, descrever uma festa de aniversário, etc).
- Dependência Conceitual, formalismo utilizado na representação de frases que utiliza um número limitado de conceitos primitivos e regras de formação.

4.3 Processamento do Discurso

Nas seções anteriores, analisamos questões referentes ao processamento sintático e à obtenção de representação semântica para frases. Essas duas etapas de processamento, contudo, analisam as frases em isolamento, não considerando os fenômenos lingüísticos que ocorrem entre frases, e requerem análise contextual para sua interpretação. Vejamos o exemplo a seguir:

Exemplo (18)

1. `Antônio quer fazer uma festa de formatura na sua casa.`
2. `Ele a limpou e arrumou ontem.`

À primeira vista, parece-nos bastante simples concluir que o pronome `a` se refere a `casa`, e não a `festa`, uma vez que sabemos que não se limpa uma festa. Para um sistema de PLN, contudo, esta conclusão não é assim tão simples, uma vez que o que nos ajuda a optar por *casa* é o nosso '*conhecimento do mundo*' (ou *senso comum*). Como é possível resolver automaticamente problemas desta natureza?

Esta pergunta tem motivado o desenvolvimento de uma variada gama de teorias lingüísticas e algoritmos computacionais cujo objetivo é representar, interpretar e gerar *discurso*, a fim de construir sistemas de PLN capazes manter um diálogo com o usuário. Esses estudos também são úteis em aplicações como tradução automática.

Aqui, o termo *discurso* é utilizado para denominar texto escrito ou falado, este último produzido na forma de monólogo ou conversação. Em outras palavras, utilizamos o termo *discurso* para nomear uma produção lingüística composta por mais de uma frase.

Esta seção pode ser vista como dividida em duas partes: teoria e prática. Inicialmente, veremos um pouco do trabalho desenvolvido dentro da Lingüística (e áreas correlatas), que contribuiu para a construção de modelos para o processamento do discurso (Análise do Discurso e Pragmática). A seguir, apresentaremos alguns modelos para o processamento automático do discurso, desenvovidos dentro da abordagem de IA.

4.3.1 Análise do Discurso e Pragmática

Como vimos no início deste texto (seções 3.2.5 e 3.2.6), essas duas áreas de trabalho têm como preocupação central o estudo do *discurso*. Portanto, não se limitam a analisar frases isoladas, e sim seqüências de frases dentro do seu contexto de ocorrência. Para tanto, levam em consideração conhecimento lingüístico e conhecimento do mundo (fatores pragmáticos) que influenciam na interpretação do discurso.

Existe uma grande variedade de trabalhos para o tratamento do discurso. Veremos a seguir duas das contribuições mais relevantes, que influenciaram muito do que foi e é feito no processamento automático do discurso: o estudo da *estrutura do discurso* (da Análise do Discurso) e os *Atos da Fala* (da Pragmática).

Atos da Fala

Como já visto anteriormente, a Pragmática estuda os *enunciados* (frases com seu significado) no contexto do discurso sob o ponto de vista dos participantes (interlocutores).

Os *Atos da Fala* são as atividades desenvolvidas pelos falantes de uma língua enquanto fazem uso dela [Searle, 1971]. Esses atos podem ser analisados sob três pontos de vista quanto à sua composição:

1. Atos Locutórios – o enunciado e sua significação literal
2. Atos Ilocutórios – a significação intencional e contextual
3. Atos Perlocutórios – o efeito de um enunciado sobre o ouvinte

Atos Ilocutórios são os atos realizados por um falante enquanto ele pronuncia uma frase. Se o falante diz, por exemplo, "A janela está aberta", isto, como ato *locutório* é uma simples observação. Porém, como ato *ilocutório*, pode ser uma ordem ou um pedido para fechar a janela. Assim, a *intenção* do falante pode ser que alguém feche a janela. Se o ouvinte, ao término da frase, fechar a janela, terá realizado um ato *perlocutório*.

Veremos a seguir os tipos de Atos da Fala, seguidos de suas possíveis realizações lingüísticas.

- Assertivos → assertiva, constatação, afirmação, negação
- Diretivos → ordem, comando, requisição
- Comissivos → promessa, garantia, juramento, aposta
- Expressivos → elogio, agradecimento, parabenização
- Declarativos → (a) encontro, indicação, declaração (b) declaração de guerra (c) estímulo, resignação, excomunhão

Alguns exemplos de atos de fala são:

- `Prometo que vou à festa.`
- `Aposto que você vai gostar do livro.`
- `Fale com o diretor.`

Dado que os Atos da Fala são as unidades conceituais básicas envolvidas na comunicação, e que são realizados pela produção de frases, eles estão em um nível mais alto de descrição, relacionado à organização do discurso, que é chamado de *nível discursivo*. Aí entram considerações de ordem semântica e pragmática. O problema é que um mesmo enunciado pode realizar atos da fala diversos, dependendo do seu contexto de uso.

Podemos então concluir que a análise automática dos Atos da Fala, que é um aspecto central na interpretação dos diálogos, é uma tarefa de difícil implementação, uma vez que a formalização de *contextos* e *intenções* dos falantes ainda é muito obscura.

Estrutura do Discurso

Assim como atribuímos estruturas sintáticas e semânticas às frases, é também possível identificar estrutura no discurso. Do mesmo modo como vemos uma frase como sendo composta por palavras, o discurso aqui é visto como formado por unidades lingüísticas que contêm uma ou mais frases – os *segmentos* do discurso.

Nosso problema agora é delimitar esses segmentos, uma vez que não dispomos de delimitadores naturais como a pontuação, no caso das frases. A idéia é que cada segmento agrupe todas as frases consecutivas que tratem do mesmo assunto (o *foco* daquele trecho de discurso). Aqui, o tema geral do discurso é denominado de *tópico*. Vejamos o exemplo (19) abaixo (inspirado em [Allen, 1995, pp 507]):

Exemplo (19)

(a) `João e Maria saíram para comprar um cortador de grama novo`

(b) `porque o deles foi roubado.`

(c) `Maria viu os homens que o roubaram.`

(d) `Ela os seguiu até o final da rua,`

(e) `mas eles fugiram num caminhão.`

(f) `Depois de procurar no shopping center,`

(g) `eles concluíram que não têm dinheiro para comprar um novo.`

(h) `A propósito, João perdeu o emprego,`

(i) `por isso ele está sem dinheiro.`

(j) `Finalmente, eles decidiram comprar um cortador usado.`

Podemos identificar aqui três segmentos de discurso:

segmento 1 – que contém as frases (a), (b), (f), (g) e (j);

segmento 2 – que contém as frases (c), (d) e (e);

segmento 3 – que contém as frases (h) e (i).

A fronteira entre os segmentos é determinada pela mudança do *foco* no discurso. Como podemos observar, o segmento (2) tem como foco "Maria ter visto a fuga dos ladrões", enquanto que o segmento (3) tem como foco o "João estar sem emprego e sem dinheiro".

Existem inúmeros trabalhos em Lingüística que investigam maneiras de segmentar o discurso. Eles investigam os princípios lingüísticos que regem a estruturação de qualquer de tipo de texto/discurso (escrito ou transcrito de um discurso falado). Alguns dos trabalhos nesta área analisam a "superfície" do texto, buscando padrões lingüísticos que podem ser considerados como *marcadores discursivos* de início e fim de foco. Outros investigam o "conteúdo semântico" do texto, em busca de identificar mudanças de foco a partir do conteúdo proposicional do discurso.

Uma das abordagens mais difundida busca determinar que relações "unem" as frases que compõem cada segmento, e quais as relações existentes entre os segmentos de um texto coerente, através do estudo de fatores de *coesão* e *coerência* presentes no texto.

Coesão e Coerência Textual

Um texto deve ser tanto coerente quanto coeso, de forma que os conceitos e relações expressos devem ser mutuamente relevantes, permitindo que se façam inferências sobre o conteúdo desse texto.

Fatores de *coesão* regem a estruturação da seqüência superficial do texto; isto é, referem-se às características da estrutura de superfície de um texto, que une partes diferentes da frase ou de

segmentos maiores do discurso – *e.g.*, a função dos pronomes de fazer referência cruzada, artigos e alguns tipos de advérbios. A coesão, portanto, trabalha mais no nível "micro-textual".

Fatores de *coerência*, por outro lado, manifestam-se mais no nível macro-textual, investigando a conexão conceitual e estruturação do sentido do texto. Busca uma análise mais profundo, envolvendo os fatores que estabelecem relações casuais, pressuposições, implicações inter-frases, e o nível argumentativo. Envolve o estudo de fatores tais como o conhecimento do mundo dos falantes de uma língua, as inferências que eles fazem e as suas crenças, e ainda o modo como a comunicação é mediada através do uso de atos da fala.

Relações de Coerência

Em um discurso *coerente*, os segmentos guardam relações retóricas entre si. Cada relação desempenha uma *função comunicativa* no discurso [Grice, 1969]. A determinação dessas relações depende do conteúdo de cada segmento.

Como exemplo deste paradigma, citamos a *Teoria da Estrutura Retórica* (Rhetorical Structure Theory) [Mann & Thompson, 1987], que identificou relações retóricas entre segmentos do discurso, algumas delas listadas abaixo. Nesta teoria, um segmento é escolhido como núcleo, e os outros segmentos são considerados satélites. As relações retórica são estabelecidas entre o núcleo e os satélites. Em alguns casos, podemos ter mais de um núcleo no mesmo discurso.

Exemplos de relações retóricas são: Motivação, Habilitação, Circunstância, Elaboração, Solução, Causa, Resultado e Propósito, Antítese e Concessão, Condição e Opção, Interpretação e Avaliação, Sumário e Reapresentaçãoe Relações Multi-núcleo – Seqüência, Contraste e Junção.

Vejamos a seguir a análise de um trecho de discurso inspirado em [Dale & Delin, 1991]:

Exemplo (20)

N: `Você deveria ver o Balé de Câmera de Los Angeles (a companhia onde eu danço) na próxima semana.`

S1: `O ingresso custa $ 7.50, exceto para a noite de estréia.`

S2: `O show traz nova coreografia, e deve ser muito interessante.`

S3: `Eu danço três peças.`

O núcleo descreve uma ação que "poderá" ser realizada pelo interlocutor: ir ao balé. Aqui, identificamos as seguintes relações:

- Habilitação (N,S1)
- Motivação (N,S2)
- Motivação (N, S3)

Essas relações podem ser usadas tanto na interpretação quanto na geração de LN.

4.3.2 Processamento Automático do Discurso

O processamento automático do discurso baseia-se em teorias que investigam maneiras de particionar o discurso em segmentos, de modo que estes reflitam as mudanças de foco ocorridas no discurso.

Podemos identificar duas correntes de trabalho que são, na realidade, complementares: (1) processamento global do discurso, que investiga a estrutura do discurso como um todo, tendo como

preocupação central a delimitação dos seus segmentos; e (2) processamento local do discurso, que investiga os fenômenos lingüísticos dentro de cada segmento.

4.3.3 Processamento Global do Discurso

Trabalhos Iniciais

Como trabalhos iniciais dentro deste paradigma, citamos [Hobbs, 1979] e [Reichman-Adar, 1984], que muito contribuíram para os trabalhos atuais na área. Ambos os autores buscavam determinar relações retóricas entre frases e/ou segmentos do discurso, a depender do conteúdo semântico e pragmático de cada frase/segmento.

Hobbs caracteriza a coerência do discurso em termos de relações binárias entre frases e o discurso precedente (outra frase ou um segmento de discurso). Muitos tipos de relações podem ser identificados, sendo grupados em duas classes: *coordenação* e *subordinação*.

Essas relações são *computáveis*, e não apenas descritivas [Hobbs, 1978]. A identificação automática dessas relações é conseguida pelo uso de mecanismos de inferência em bases de conhecimento, que são modelos do mundo. A estrutura do discurso, portanto, baseia-se nas relações encontradas no discurso.

No exemplo (19), seção 4.3.1, os segmentos 2 e 3 são subordinados ao segmento 1, enquanto que as frases (a,b), (f,g) e(k) são coordenadas (estão no mesmo nível de discurso).

A *Teoria do Espaço de Contexto* (Context Space Theory) [Reichman-Adar, 1984] descreve um modelo para interpretar e gerar discurso. Este modelo utiliza a teoria dos Atos da Fala na identificação das *intenções* dos falantes, a fim de auxiliar no processo de análise do discurso. Uma base de conhecimento baseada em Redes de Transição Aumentadas armazena *regras de discurso* derivadas da teoria dos Atos da Fala [Grice, 1969].

Nesta teoria, o discurso é particionado em *espaços de contexto* (segmentos), de tal modo que cada foco do discurso tem um espaço de contexto associado. Espaços de contexto são esquemas que guardam, por exemplo: uma representação das frases contidas naquele espaço, ligações para os espaços de contexto co-relacionados, juntamente com a especificação das relações entre eles, assinalamentos de nível de foco dos elementos do discurso em cada espaço, etc.

Os espaços de contexto podem ser vistos como uma maneira de representar informação relevante sobre cada segmento do discurso. O nível de foco dos elementos do discurso especifica a importância de cada elemento no enunciado do falante. Esse nível auxilia, por exemplo, na determinação de antecedentes de referências pronominais – onde os elementos de nível mais alto são, normalmente, substituídos por pronomes.

Esses dois trabalhos exerceram grande influência na pesquisa posterior na área de processamento do discurso, tanto no nível global quanto local.

A Teoria da Estrutura do Discurso

A Teoria da Estrutura do Discurso [Grosz & Sidner, 1986] é talvez o modelo computacional mais completo para processamento de discurso. Aqui, a estrutura do discurso é vista como formada por três constituintes interativos, cada um lidando com um aspecto diferente do enunciado no discurso: a estrutura lingüística, a estrutura intencional, e o estado de atenção.

A *estrutura lingüística* consiste nos segmentos do discurso e nas relações entre eles. Neste nível de representação, apenas uma relação é considerada, chamada de *encaixe (embedding)*. Aqui, um segmento pode 'embutir' um ou mais segmentos, algo semelhante à relação de subordinação de Hobbs.

Esta relação delimita segmentos, e é identificada, entre outras coisas, com base em marcadores discursivos, que são expressões lingüísticas como: *A propósito, Bem, Em resumo, etc.* Outras dicas a considerar são o uso de pronomes (que, em sua maioria, limitam-se a referenciar elementos dentro de um mesmo segmento), tempo verbal, e intonação (quando se trata de discurso falado).

A *estrutura intencional* contém os propósitos (intenções) expressados por cada segmento. Esta teoria identifica um propósito mais relevante para o discurso como um todo, chamado de *propósito do discurso* (PD), e os propósitos de cada segmento (*propósito do segmento do discurso* (PSD)), onde cada PSD contribui para o discurso alcançar seu PD geral.

Devido à imensa variedade de intenções que podem servir como PD e PSD, a estrutura intencional não parte de um conjunto pré-determinado de relações de coerência, como ocorre no trabalho de Hobbs, Reichmann e Mann & Thompson, por exemplo. Aqui, são consideradas apenas duas relações entre PSDs: *domínio* (dominance) e *satisfação-precede* (satisfaction-precedence). A relação de domínio estabelece uma hierarquia entre os vários PSDs no discurso, semelhante à relação de suborninação em Hobbs. Ainda, PSD_i satisfação-precede PSD_j quando PSD_i deve ser satisfeito antes de PSD_j (isto é, os propósitos do segmento i devem ser satisfeitos antes dos propósitos do segmento j).

Em resumo, a estrutura intencional consiste no PD, nos PSDs dos segmentos do discurso, e nas relações entre esses PSDs. O reconhecimento dessas relações pode ser conseguido via reconhecimento de "planos" (ou intenções), estudos desenvovidos na área de Planejamento (*Planning*), em IA [Allen, 1983]. O estado de atenção, descrito a seguir, também colabora para a identificação dessas relações.

O *estado de atenção* reflete o foco de atenção de cada segmento do discurso. Este constituinte, assim como as outras duas estruturas acima, não existe a priori, sendo formado à medida que o discurso evolui.

Este estado consiste em uma pilha de *espaços de foco* (conceito semelhante aos *espaços de contexto* [Reichman-Adar, 1984]), onde cada segmento do discurso tem seu espaço de foco associado. Cada espaço contém uma representação das entidades que estão em evidência (em foco) naquele segmento do discurso. Esta representação inclui as propriedades e relações entre as entidades representadas, assim como o PSD do segmento.

À medida que o discurso vai sendo processado, espaços de foco são inseridos ou removidos da pilha, com base nas relações de domínio e satisfação-precede. Um espaço só é retirado da pilha quando seu PSD for satisfeito.

Ao final do processamento do discurso, a pilha de focos estará vazia, enquanto que a hierarquia de domínio estará totalmente construída.

Vejamos a seguir um exemplo da teoria em uso, como apresentado em [Grosz & Sidner, 1986, pp 186–193]:

Exemplo (21) Um fragmento de discurso.

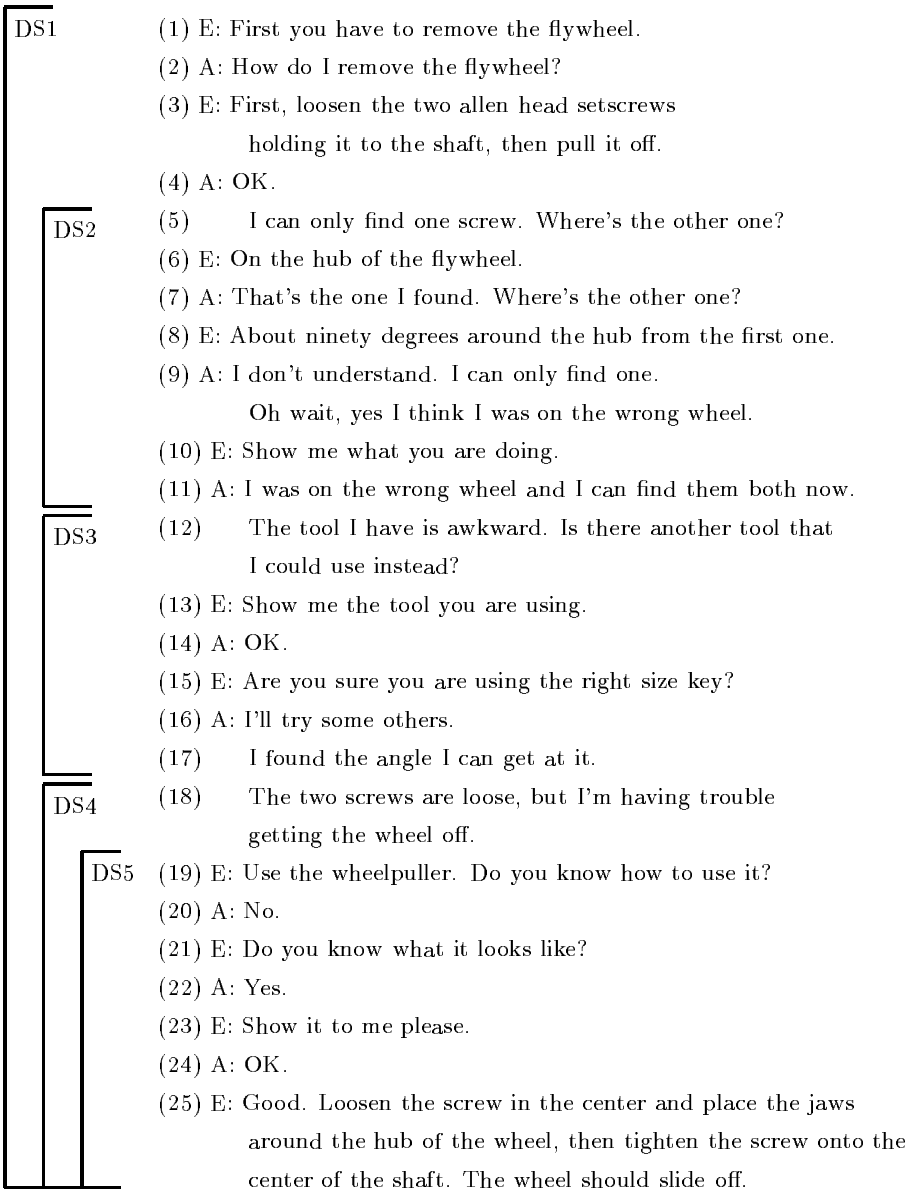

Figura 11: Estrutura Lingüística.

Primary Intentions:

I1: Intend E_xpert (Intend A_pprentice (Remove A flywheel)))
I2: (Intend A (Intend E (Tell E A (Location other setscrew))))
I3: (Intend A (Intend E (Identify E A another tool)))
I4: (Intend A (Intend E (Tell E A (How (Getoff A wheel)))))
I5: (Intend E (Know-How-to A (Use A wheelpuller)))

Dominance Relationships:

I1 DOM I2
I1 DOM I3
I1 DOM I4
I4 DOM I5

Satisfaction-Precedence Relationships:

I2 SP I3
I2 SP I4
I3 SP I4

Figura 12: Estrutura Intencional.

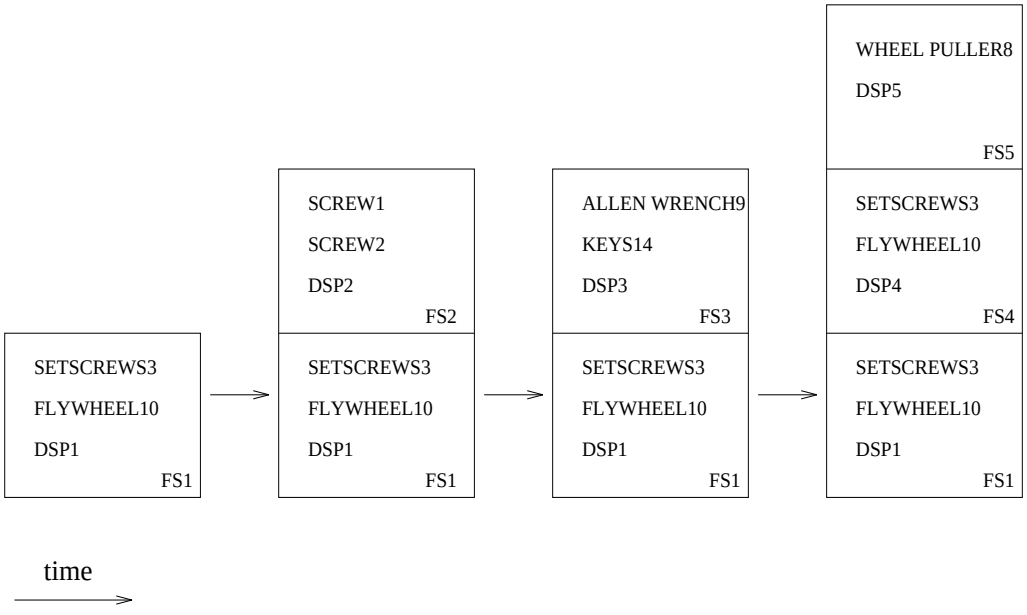

Figura 13: Estado de Atenção.

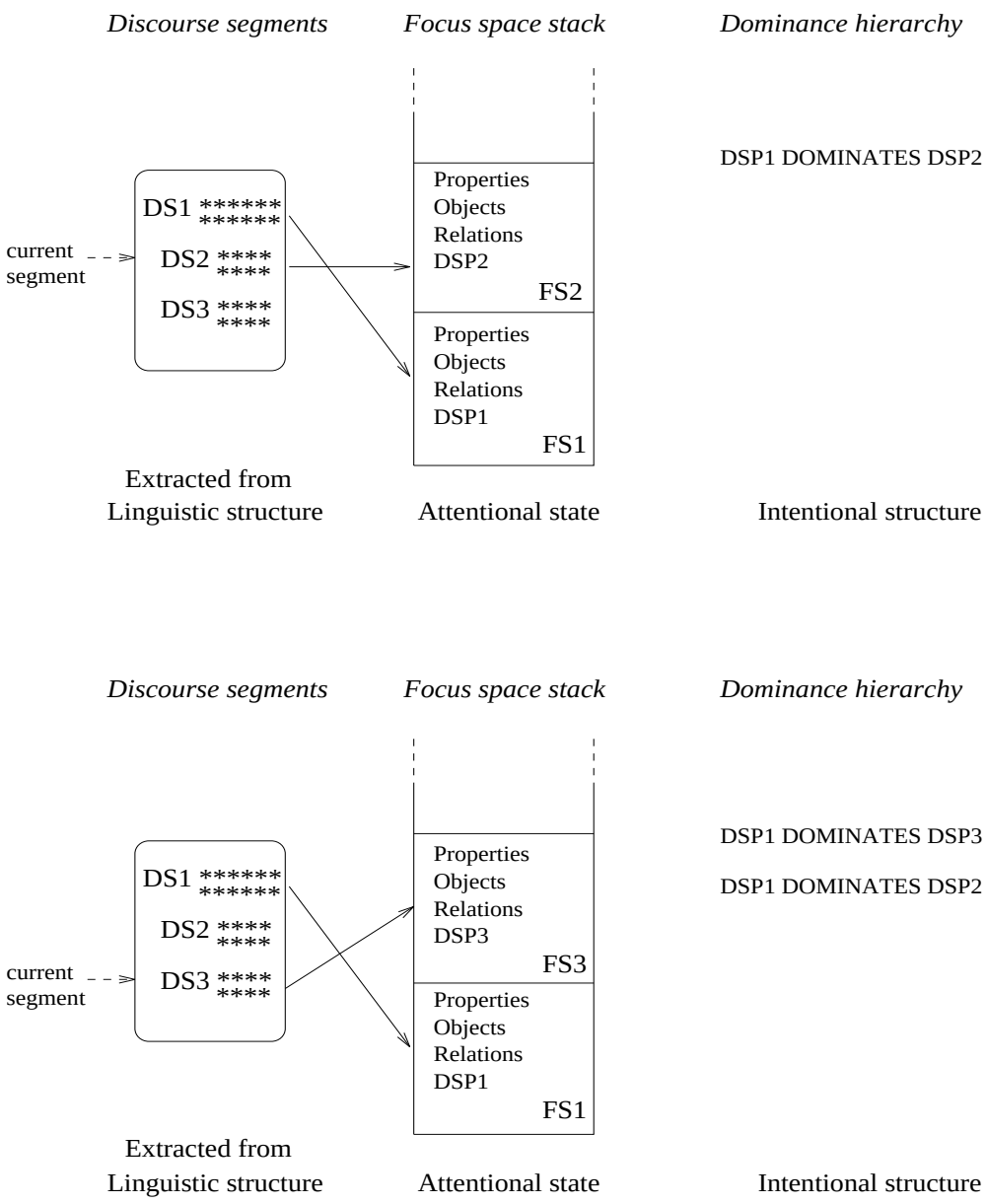

Figura 14: Processamento do Discurso.

O grande problema aqui ainda é a identificação das fronteiras de cada segmento, uma vez que cada um dos constituintes desta teoria simultaneamente depende da informação dos outros dois constituintes para processar o discurso. Vale ressaltar que não existe uma implementação completa deste modelo, havendo apenas implementações de algumas de suas idéias básicas.

4.3.4 Processamento Local do Discurso

Esses trabalhos assumem que o discurso já está segmentado, concentrando-se em estudar as mudanças de foco dentro de cada segmento como um fator essencial para a determinação de referentes de pronomes. Entre as contribuições mais significativas, citamos Hobbs [Hobbs, 1978], [Webber, 1981], e a Teoria de Centros (*Centering Theory*) [Grosz *et al.*, 1983] e Sidner [Sidner, 1983].

Trabalho de Sidner sobre Foco

Destacamos aqui o trabalho de Sidner [Sidner, 1983], que deu uma grande contribuição para o problema da resolução automática de referência pronominal. O algoritmo desenvolvido por Sidner utiliza uma base de conhecimento onde as entidades do discurso estão previamente representadas, juntamente com características que auxiliam na resolução de referências pronominais.

O processo de resolução pronominal consiste em procurar o *foco do discurso* dentre os elementos do discurso. Um *foco previsto* é selecionado depois do processamento da primeira frase, com base em preferências sintáticas e semânticas (*e.g.*, se o elemento é o sujeito da frase, ou se é o tema de um verbo). Com a continuação do processamento, este foco é confirmado ou rejeitado, pela análise dos pronomes encontrados no discurso. Quando uma previsão incorreta é detectada, um novo foco previsto será escolhido dentre os candidatos disponíveis.

Além do foco do discurso, esse processo também determina o *foco ator*, que é um elemento do discurso que assume o papel de agente em algum evento. O foco ator só pode se tornar o foco do discurso quando não há outro elemento disponível para tal. Esta distinção é necessária porque ambos o foco do discurso e o foco ator podem ser referenciados por pronomes ao mesmo tempo.

Vejamos um exemplo de funcionamento deste processo:

Exemplo (22)

1. `Antônio quer fazer uma grande festa de formatura na sua casa.`
2. *Ele a* `limpou e arrumou ontem,`
3. `de modo a poder receber todos os seus amigos.`

O objetivo aqui é determinar quem é o referente dos pronomes *Ele* e *a* na frase (2). Vejamos como os focos são escolhidos neste exemplo:

- *Foco ator* = `Antônio`
- *Foco esperado 1* = `festa de formatura`
- *Foco esperado 2* = `casa de Antônio`
- *Foco do discurso* = `casa de Antônio`

O *foco esperado 1* (`festa de formatura`) é rejeitado quando o pronome *a* é encontrado na segunda frase. Esta substituição é feita com base num processo de inferência na base de conhecimento do sistema. O resultado do processamento é ligar o pronome *Ele* ao foco ator (`Antônio`), e o pronome *a* ao foco do discurso (`casa de Antônio`).

Qualquer novo elemento do discurso é tratado como um *foco potencial*, sendo armazenado em uma lista de *foci potenciais*. Se o próximo pronome encontrado se referir a algum foco potencial, este foco se tornará o foco do discurso. A lista de foci potenciais sobrevive apenas durante o processamento da frase que os introduziu. No caso do exemplo acima, `casa de Antônio` foi selecionado como foco potencial antes de se tornar foco esperado, e posteriormente foco do discurso.

Este algoritmo também prevê a possibilidade de uma mudança de foco com posterior volta a algum foco anterior. O algoritmo guarda todos os elementos que foram foco do discurso na *pilha de foci*, de modo que eles poderão ser "recuperados" caso o discurso volte a qualquer um deles.

4.3.5 Considerações Finais

Muito ainda resta para ser dito sobre processamento do discurso. Contudo, devido aos limites de tempo e espaço, deixamos apenas algumas indicações que poderão ser úteis a quem estiver interessado em investigar mais de perto o tema.

Entre os trabalhos para processamento local do discurso, destacamos a teoria dos Centros (*Centering*) [Grosz *et al.*, 1983], e a Teoria de Representação do Discurso *Discourse Representation Theory* [Kamp & Reyle, 1993], que também têm como preocupação central a resolução de referências pronominais.

No que concerne a segmentação do discurso, além dos trabalhos iniciais vistos acima [Mann & Thompson, 1987], destacamos aqui [Grosz, 1977] e [Grosz & Sidner, 1986].

5 Geração de Linguagem Natural

5.1 Introdução

O processo Geração de Linguagem Natural (GLN), é essencialmente o inverso do processo da interpretação. Um gerador de linguagem natural é um software que recebe como entrada um conjunto de *elementos de conteúdo* (informações) e/ou de *objetivos de comunicação*, e que produz (automaticamente) como saída, uma frase ou um texto que:

- Satisfaz os objetivos de comunicação;
- Expressa os elementos de conteúdo relevantes;
- É lingüísticamente correto;
- É composto dinamicamente, usando um léxico e uma gramática de uma língua dada.

5.1.1 Entrada e Saída de um Gerador de Linguagem Natural

A representação dos elementos de conteúdo guarda estreita relação com a aplicação para a qual o gerador foi desenvolvido, podendo ser:

- Grandes tabelas estatísticas;
- O rastreamento do raciocínio de um sistema especialista ou de um simulador;
- A resposta a uma consulta a um banco de dados.
- Uma fórmula lógica;
- Uma rede semântica (usada, por exemplo, como representação interlíngua em um tradutor automático multilíngüe).

Os objetivos de comunicação também dependem intimamente da aplicação, podendo ser:

- Resumir as informações fornecidas como entrada;

- Induzir o usuário (*i.e.*, o leitor do texto gerado como saída) executar alguma ação[7];
- Respeitar regras de estilo (por exemplo, para a redação automática de documentação técnica).

Em alguns sistemas, os objetivos de comunicação não são explicitamente dados como entrada. Nestes casos, os objetivos estão embutidos na arquitetura do gerador, que recebe como entrada apenas algum conteúdo a expressar. Este é o caso do sistema ANA [Kukich, 1988], que gera relatórios diários sobre as atividades da Bolsa de Valores, num estilo jornalístico. A única entrada desse sistema consiste em valores numéricos de um conjunto de indicadores financieros a cada hora. O objetivo de comunicação de ANA é único e implícito: resumir, em um parágrafo conciso, as flutuações de indicadores em destaque no dia. Um exemplo de entrada e saída do sistema é dado na Fig. 15.

Volume	Advances	Declines	Unchanged
55,860K	772	660	417

Time	Indicator	Size	Value
10am	Industrial	30	814.22
10am	Transportation	20	318.05
10am	Utilities	15	106.39
10am	Stocks	65	315.79
11am	Industrial	30	813.55
11am	Transportation	20	317.60
11am	Utilities	15	106.22
11am	Stocks	65	315.46
12pm	Industrial	30	814.12
12pm	Transportation	20	317.82
12pm	Utilities	15	106.57
12pm	Stocks	65	315.81
2pm	Industrial	30	814.12
2pm	Transportation	20	317.60
2pm	Utilities	15	106.78
2pm	Stocks	65	315.85
3pm	Industrial	30	810.12
3pm	Transportation	20	316.47
3pm	Utilities	15	106.70
3pm	Stocks	65	314.63
4pm	Industrial	30	810.14
4pm	Transportation	20	317.00
4pm	Utilities	15	106.83
4pm	Stocks	65	314.90

Thursday, June 24 1982 – Wall Street's securities markets meandered upward through most of the morning, before being pushed downhill late in the day yesterday. The stock market closed out the day with a small loss and turned to mixed showing in moderate trading.
The Dow Jones average of 30 industrials declined slightly, finishing the day at 810.41, off 2.76 points. The transportation and utility indicators edged higher. Volume on the Big Board was 55,860K shares compared with 62,710K shares on Wedsnesday. Advances were ahead by about 8 to 7 at the final bell.

Figura 15: Exemplo de Entrada/Saída de um gerador de texto.

[7] Em termos da teoria dos atos de fala apresentada na seção 4.3.1, um tal objetivo de comunicação é de natureza *perlocutório*.

Em outros sistemas, os objetivos de comunicação constituem a única entrada do gerador. Nestes casos, o gerador fica responsável pela busca, em uma base de dados ou de conhecimento, dos elementos de conteúdo relevantes para satisfazer os objetivos de entrada. Este é o caso do sistema ADVISOR-II [Elhadad *et al.*, 1997], que gera conselhos para um aluno sobre a organização da programação de cursos. Um exemplo de objetivo de comunicação que ADVISOR-II recebe como entrada é o de convencer o aluno a cursar ou a não cursar determinada disciplina em um dado semestre. Para satisfazer esse objetivo, ADVISOR-II busca fatos em uma base de conhecimento que especifica as características de cada disciplina (se ela é obrigatória, quais são seus pré-requisitos, quem a ensina quando, etc.). A figura 16 mostra duas respostas à mesma pergunta: *"Should I take AI?"* ("Eu deveria cursar IA?"), de um aluno que gosta de programação e não gosta de matemática. ADVISOR-II gera as respostas a partir da mesma base de conhecimento, porém com objetivos de comunicação opostos.

`Objetivo de comunicação = take(reader,AI)`:
`Texto de saída` = "AI covers many interesting topics such as NLP, Vision and Expert Systems. And it involves a good amount of programming. So it should be interesting."

`Objetivo de comunicação = not(take(reader,AI))`:
`Texto de saída` = "AI covers logic, a very theoretical topic, and it requires many assignments. So it could be difficult."

Figura 16: Exemplo da influência dos objetivos de comunicação na saída de um gerador.

5.1.2 Geração *vs.* Interpretação

Um gerador de linguagem natural incorpora técnicas de IA (especialmente na representação do conhecimento, na busca heurística e planejamento) e de Lingüística Computacional (em particular, as gramáticas formais).

Um gerador se distingue de um interpretador pelas seguintes características:

- O gerador tem controle sobre a variedade de formas lingüísticas usadas para expressar os conceitos do domínio de aplicação (uma vez que elas constituem a saída, e não a entrada, do sistema).
- A gramática deve partir da *função* de cada elemento lingüístico (palavra, constituinte sintático, oração) do texto, ao invés de partir da sua estrutura.
- O léxico deve partir dos *conceitos* do domínio de aplicação, ao invés de partir das suas palavras.
- O gerador é responsável pela seleção e pela organização do conteúdo do texto, tarefas sem equivalente em um interpretador, uma vez que são efetuadas por quem escreveu o texto de entrada.

5.1.3 Geração *vs.* Textos Pré-fabricados

Um gerador se distingue de um sistema baseado em textos pré-fabricados pelas seguintes características:

- O gerador compõe o texto *dinamicamente* e *composicionalmente*, a partir de um léxico e de uma gramática da língua, implementados em um formalismo computacional (de preferência, declarativo).[8].

[8] Na prática, os léxicos implementados são de cobertura limitada à sub-língua do domínio de aplicação do gerador. Por exemplo, ANA cobre somente o vocabulário financeiro usado em relatórios da bolsa de valores.

- O gerador tem uma arquitetura *modular*, onde os pares
  < elementos de conteúdo , objetivos de comunicação >
  são mapeados em textos *indiretamente*, através de algumas representações intermediárias (como aquelas mostradas na Fig. 1).

Conseqüentemente:

- Aumentar a cobertura de um gerador *não* requer codificar uma lista combinatorialmente explosiva de mapeamentos entre pares < elementos de conteúdo , objetivos de comunicação > e textos. Pode-se apenas aumentar o léxico com algumas entradas e adicionar algumas regras à gramática, e então testar as interações desses novos elementos de conhecimento com os já existentes.
- Um gerador pode delegar parte de suas tarefas a componentes especializados, como por exemplo, uma gramática computacional abrangente de uma língua dada, que pode ser re-utilizada em domínios de aplicação totalmente diferentes.

5.1.4 As Tarefas de um Gerador

Um sistema completo para geração de linguagem natural deve efetuar as seguintes tarefas e subtarefas:

1. Determinação do conteúdo → o que dizer?
2. Organização do conteúdo → quando dizer o quê?
   (a) Organização do discurso
   (b) Organização das frases
3. Realização (lingüística) do conteúdo → como dizê-lo?
   (a) Lexicalização (*i.e.,* escolha de palavras de classes abertas – verbos, substantivos, adjetivos e advérbios – e de estruturas temáticas – *e.g,* agente, paciente, instrumento).
   (b) Realização sintática
      i. Sintaxe de superfície (*e.g.,* concordância, mapeamento dos papeis temáticos para os elementos de superfície da frase como sujeito, objeto, complemento, adjunto).
      ii. Palavras de classes fechadas (pronomes, artigos, conjunções, etc.)
      iii. Morfologia
      iv. Linearização

A *determinação do conteúdo* consiste em selecionar os elementos de conteúdo relevantes para satisfazer os objetivos de comunicação. Essa tarefa corresponde essencialmente aos tratamentos pragmático e semântico na Fig. 1. Por exemplo, como ilustrado na Fig. 16, o gerador ADVISOR-II decidiu mencionar o fato de que o curso de IA requer programação, a fim de convencer o aluno a cursá-lo. Para desencorajar o aluno a cursar IA, o sistema prefere mencionar o fato que o curso de IA também inclui lógica.

A *organização do conteúdo* consiste em agrupar os elementos de conteúdo selecionados em unidades lingüísticas, definindo também as dependências hierárquicas e lineares entre essas unidades dentro de uma estrutura maior. A organização do conteúdo se decompõe em dois níveis: (1) a organização do discurso (que corresponde ao tratamento do discurso na Fig. 1), onde a unidade lingüística é a oração e a estrura maior é o texto, e (2) a organização das frases, onde a unidade lingüística é o constituinte sintático e a estrutura maior é a oração. No exemplo ilustrado na Fig. 15, o gerador ANA decidiu, no nível de processamento do discurso, agrupar o valor final e a variação do indicador industrial na terceira oração do texto, enquanto as variações dos indicadores dos transportes e das "*utilities*". foram agrupados na quarta oração. O sistema ainda decidiu expressar,

na terceira oração, o valor do indicador industrial, depois de sua variação, como um constituinte dependente.

A *lexicalização* consiste em selecionar as palavras raízes e as estruturas temáticas para expressar cada elemento de conteúdo (no contexto da estrutura lingüística na qual ele foi alocado). Isto corresponde a uma parte do tratamento semântico mostrado na Fig. 1. Por exemplo, como ilustrado na Fig. 15, o gerador ANA decidiu resumir as variação do indicador das "*utilities*" usando o verbo *"to edge"* e o adjetivo *"higher"*, dentro de uma estrutura temática onde o indicador é o agente, e *"higher"* designa a locação resultante de sua ação.

A *realização sintática* é a etapa final da geração de um texto, e inclui os tratamentos sintático e morfológico da Fig. 1. Esta étapa sub-divide-se em:

1. Mapear a estrutura temática de cada frase em uma estrutura sintática de superfície (*e.g.*, na quarta frase gerada por ANA (Fig. 15), mapear o agente no sujeito e a locação no complemento do sujeito).
2. Aplicar as regras de sintaxe, como concordância entre o sujeito e o verbo.
3. Escolher as palavras de classes fechadas (*e.g.*, *"many"*, *"such as"* e *"and"* na primeira frase gerada por ADVISOR-II – Fig. 16).
4. Flexionar as palavras raízes de classes abertas (*e.g.* flexionar o verbo *"to edge"* para sua forma no passado *"edged"*, na quarta frase gerada por ANA – Fig. 15).
5. Linearizar a árvore sintática em uma cadeia de palavras s, atravessando essa árvore em profundidade e da esquerda para direita.

Cada um dos geradores existentes se concentra em um sub-conjunto diferente das tarefas listadas acima, implementando-as em arquiteturas com vários graus de modulariedade. Um gerador completo, com a mais simples das arquiteturas plenamente modulares, teria um componente especializado para cada uma dessas tarefas, e esses componentes seriam organizados em uma estrutura de níveis onde o fluxo de informação seria unidirectional, na seguinte ordem: (1) determinador do conteúdo, (2) planejador do discurso, (3) planejador das frases, (4) lexicalizador, e (5) realizador sintático. Em tal estrutura, cada componente recebe sua entrada do componente anterior a ele, e passa sua saída para o componente seguinte.

Apresentamos-nos, a seguir, alguns detalhes sobre a tarefa de realização lingüística, antes de concluir com tópicos avançados de pesquisa na área. Detalhes sobre as tarefas de determinação e organização do conteúdo são encontrados em [McKeown & Swartout, 1987] que traz uma apresentação didática dos problemas e das técnicas usadas para essas tarefas. Estudos clássicos dessas tarefas incluem: [Kukich, 1983] [McKeown, 1985] [Appelt, 1985] [Hovy, 1991] [Dale, 1992] [Carcagno & Iordanskaja, 1993].

5.2 Realização Sintática

5.2.1 Entrada/Saída do Realizador Sintático

A entrada típica de um componente de realização sintática é uma árvore temática parcialmente lexicalizada que especifica, para cada constituinte sintático: seu papel temático (*e.g*, agente, paciente, instrumento), sua categoria sintática, suas palavras de classes abertas, seus traços sintáticos não-default[9], e alguns traços não-sintáticos provenientes do componente de tratamento pragmático e/ou do componente de tratamento do discurso.

[9] Em IA, o valor "default" de um atributo (ou variável) de uma dada categoria é seu valor mais comum (ou assumido) na ausência de mais conhecimento. Por exemplo, o valor "default" do atributo `pode-voar` para a categoria `ave` é `sim`. No atual contexto lingüístico, o valor "default" do atributo `voz` para a categoria `cláusula` é `ativa`.

Por exemplo, a Fig. 17 mostra a entrada G_1^e para gerar a frase O_1 *"Ela entrega o rascunho ao editor"* (no formalismo das gramáticas de unificação funcional introduzido na secção 5.2.3).

Especificação de entrada G_1^e:

$$\begin{bmatrix} categoria & oracao \\ processo & \boxed{0}\begin{bmatrix} tipo & composto \\ relacao & posse \\ palavra-raiz & \text{“}entregar\text{”} \end{bmatrix} \\ papeis-tematicos & \begin{bmatrix} agente & \boxed{1}\begin{bmatrix} categoria & pronome \\ genero & feminino \end{bmatrix} \\ beneficiario & \boxed{2}\begin{bmatrix} categoria & SN \\ palavra-raiz & \text{“}editor\text{”} \end{bmatrix} \\ possuidor & \boxed{2} \\ possuido & \boxed{3}\begin{bmatrix} categoria & SN \\ palavra-raiz & \text{“}rascunho\text{”} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Frase de saída F_1: *"Ela entrega o rascunho ao editor."*

Figura 17: Exemplo de entrada para o realizador sintático.

5.2.2 Sub-tarefas do Realizador Sintático

Para gerar uma frase a partir de uma entrada dada, o realizador sintático tem que efetuar, além das cinco sub-tarefas já mencionadas na seção 5.1.4, as seguintes sub-tarefas:

1. Fornecer valores default para os traços não mencionado na entrada, (*e.g.*, por exemplo, os sub-traços `[tempo presente]` e `[voz ativa]` do atributo `processo`).
2. Fornecer uma ordem parcial entre constituintes
   (*e.g.*, `sujeito > verbo > objeto-direito > objeto-indireito > adjunto`
   no nível mais alto na estrutura de F_1).
3. Controlar a paráfrase sintática a partir dos traços discursivos ou pragmáticos (*e.g.*, se a entrada da Fig. 17 fosse aumentada com o traço `[enfoque` $\boxed{2}$ `]`, que requer enfatiza-lo participante `possuído`, o realizador deveria escolher uma forma alternativa que colocasse esse papel temático na frente[10]; conseqüentemente, o realizador sintático escolheria a voz passiva, que mapeia o papel temático `possuído` no papel sintático de frente, o `sujeito`, gerando assim:
   (F_2) *"O rascunho é entregue ao editor por ela."*

5.2.3 Gramáticas de Unificação Funcional

As Gramáticas de Unificação Funcional (GUFs) [Kay, 1979] [Elhadad, 1993a] são um formalismo muito usado na implementação de realizadores sintáticos para geração de linguagem natural (*e.g.*, sistemas TEXT [McKeown, 1985], KAMP [Appelt, 1985], COMET [McKeown *et al.*, 1990], COOK [Smadja & McKeown, 1991] EPICURE [Dale, 1992] TAILOR [Paris, 1993], PLANDOC [Kukich *et al.*, 1994], KNIGHT [Lester, 1994] FLOWDOC [Passoneau *et al.*, 1996] STREAK [Robin & McKeown, 1996] ADVISOR-II [Elhadad *et al.*, 1997], SUMMONS [McKeown & Radev, 1995] e MAGIC [Dalal *et al.*, 1996]).

As GUFs baseiam-se nos seguintes conceitos básicos:

[10] A estratégia default para enfatizar um constituinte é de o colocar no início da oração.

- Representação uniforme da entrada e da saída como estruturas recursivas de traços (como no caso da Fig. 17) – chamadas de *Descrições Funcionais* (DFs).
- Representação do mesmo programa com DFs de um tipo especial – chamadas de *Gramática Funcional* (GF) – que incluem disjunções (introduzindo a necessidade de busca), e anotações de controle (permitindo codificar heurísticas para guiar essa busca).
- Representação das funções (papéis temáticos e sintáticos) de cada constituinte através dos *nomes* dos traços (*e.g.*, `possuído` para "o rascunho" em G_1^e).
- Representação das dependências estruturais entre os constituintes lingüísticos, através da estrutura recursiva das DFs e do traço chave `categoria` (*e.g.*, em G_1^e, a DF para "o rascunho" aparece embutida no seu papel temático `possuído`, e inclui o traço `<categoria SN>`).
- Computação baseada na operação recursiva e monotônica de unificação *funcional* da DF de entrada com a GF (programa), resultando na DF de saída.

Semelhante à unificação posicional do PROLOG, a unificação funcional usa uma busca default em profundidade (depth-first) com backtracking built-in para a resolução das disjunções. Além disso, ela também usa uma busca em largura (breadth-first) para a recursão sobre os constituintes lingüísticos. A DF de entrada completa é unificada com a GF uma primeira vez. O resultado é o nível mais alto da DF de saída. Em seguida, cada sub-constituinte lingüístico nesse nível é individualmente unificado com a GF. A DF de saída fica completa quando todos os sub-constituintes de mais baixo nível foram unificados com a GF.

Outras diferenças essenciais entre a unificação funcional e a de PROLOG são as seguintes:

- Uma DF não tem um número de atributos definido; conseqüentemente ela codifica, inerentemente, descrições *parciais* de uma entidade, exatamente como as frases de linguagem natural.
- Em uma DF, os atributos não são ordenados, mas diferenciados por nomes, independentemente de sua posição, liberando assim o programador da tarefa de se lembrar da posição associada a cada atributo.

No paradigma das GUFs, todas as sub-tarefas da realização sintática são realizadas através da unificação da DF de entrada com a GF, exceto a linearização[11]. Esta última sub-tarefa é um processo separado, que atravessa, de uma maneira top-down, a DF de saída obtida pela unificação, e imprime as palavras flexionadas na ordem indicada nos atributos especiais `padrão`.

Para ilustrar ambos (1) a recursão em largura sobre constituintes e (2) a linearização, considere o resultado G_1^s (Fig. 18), da unificação de G_1^e com uma micro-GF simplificida do portugûes G^p, (não mostrada aqui por falta de espaço). Porque a unificação é uma operação monotônica, todos os traços da entrada G_1^e ficam em G_1^s. A diferença entre as duas reside nos traços que provêem de G^p e completam a descrição parcial de entrada G_1^e durante a unificação. Eles são enfatizados em negrito em G_1^s.

A DF corrente G^c é inicializada com G_1^e para a primeira unificação. Essa unificação resulta no enriquecimento de G^c com os traços em negrito, no mais alto nível em G_1^s:

- `papeis-sintaticos`, `voz` e `modo`, que indicam o mapeamento entre a estrutura semântica de entrada e a estrutura sintáctica;
- `padrão`, que indica a ordem linear desses papéis sintáticos na frase de saída.

A seguir, cada sub-DF de G^c que corresponda a um sub-constituinte é recursivamente unificada com a GF G^p: respectivamente $\boxed{1}$, $\boxed{2}$ e $\boxed{3}$. Essas unificações embutidas de nível 1 resultam no enriquecimento de $\boxed{1}$, $\boxed{2}$ e $\boxed{3}$ com seus traços em negrito em G_1^s. Note que, no caso de $\boxed{2}$, a

[11] E a tarefa da morfologia, no casos das línguas com morfologia simples, como o Inglês.

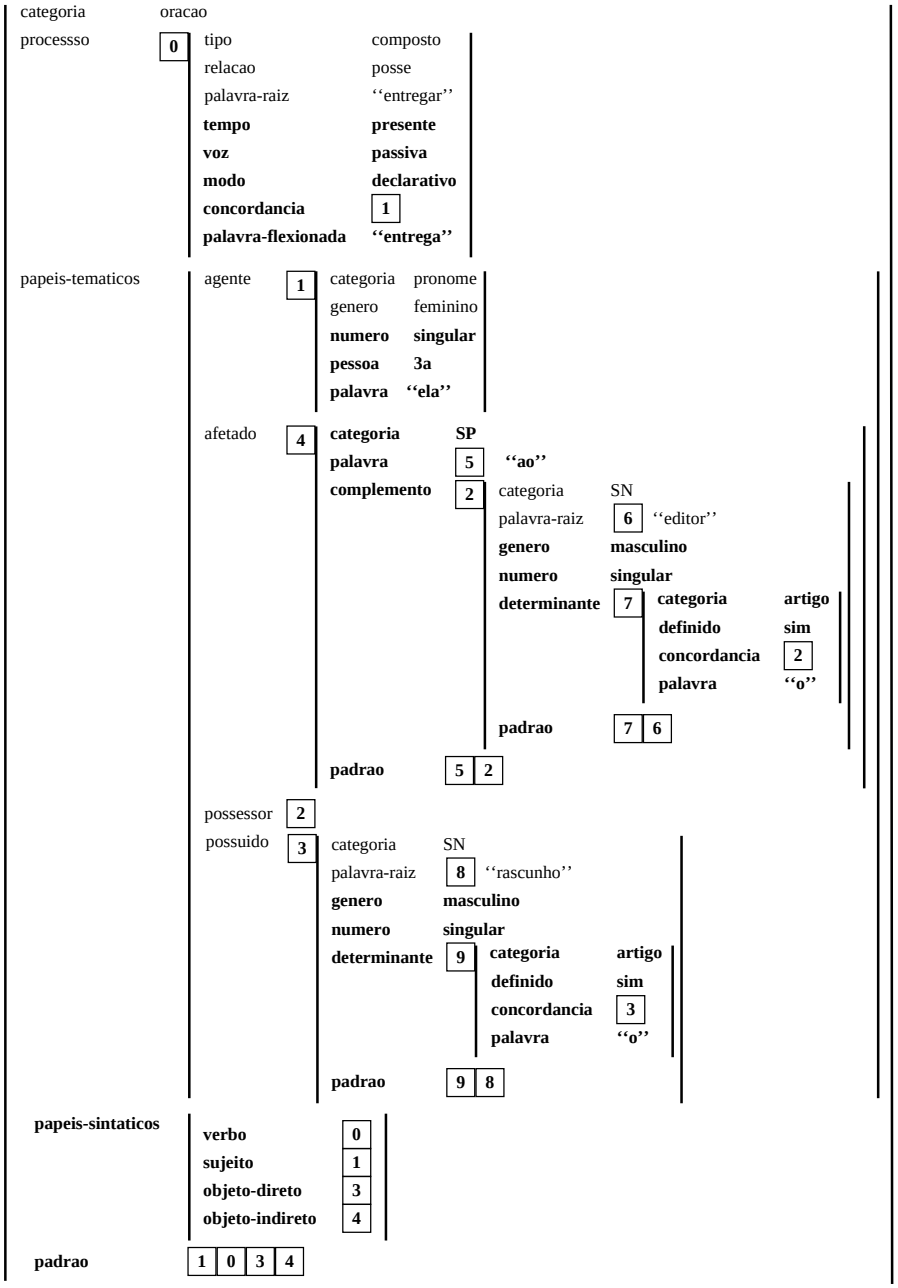

Figura 18: G_1^s obtida pela unifação de G_1^e com uma micro-gramática do portugûes.

unificação também resulta na criação de um novo constituinte, [4], dentro do qual [2] fica embutido em G_1^s. As unificações de ambos [2] e [3] também desparam mais duas unificações, de [7] e [9] respectivamente, cada uma para especificar o determinante do sintagma nominal correspondente. Como essas duas unificações embutidas de nível 2 enriquecem G^c apenas com atributos atômicos, a recursão sobre constituintes termina. Qual tipo de constituinte desencadeia recursão com a GF é indicado declarativamente na mesma GF para cada categoria sintática.

O processamento morfológico consiste em: acrescentar as DF folhas dos constituentes de classe lexical abertas do traço `palavra-flexionada` e acresentar os constituentes de classe lexical fechada do traço `palavra`, ambos sobre as restrições especificadas pelo traços de concordância.

Uma vez que G_1^s é produzida pelo processo de unificação recursivo que nós acabamos de seguir, ela é linearizada na frase F_1. A linearização consiste em atravessar a estrutura de constituintes de G_1^s em profundidade:

1. Seguindo o traço `padrão` dos constituintes não folhas;
2. Imprimindo o valor do traço `palavra-flexionada`, `palavra` ou `palavra-raiz`[12] dos constituintes folhas.

Esse processo resulta no refinamento gradual das ordens parciais entre constituintes, indicados nos traços `padrão` na ordem total entre as palavra da oração. No exemplo de G_1^s, as etapas sucessivas desse refinamento são as seguintes:

1. {1,0,3,4}
2. {1,0,9,8,5,2} (3 disparando 9 e 8, 4 disparando 5 e 2)
3. {1,0,9,8,5,7,6} (2 disparando 7 e 6)

5.3 Lexicalização

5.3.1 Entrada/Saída do Lexicalizador

A saída do lexicalizador são as árvores temáticas lexicalizadas que o realizador aceita como entrada, cujo exemplo é mostrado na Fig. 17. Ainda não existe entrada realmente padrão para o lexicalizador, como é o caso para o realizador sintático.

Apesar disso, no sentido mais estrito da lexicalização, a entrada do lexicalizador para a geração de uma frase pode ser definida como uma árvore semântica onde os elementos de conteúdo já estão agrupados hierarquicamente em uma estrutura que prefigura a estrutura dos constituintes lingüísticos, mas sem especificar nenhuma palavra ou forma sintática específica.

Os arcos dessa árvore são relações retóricas (*e.g.*, como aquelas da Teoria das Estruturas Retóricas apresentada na seção 4.3.1), ou temáticas (*e.g.*, como aqueles dos padrões de casos apresentados na seção 4.2.2) entre os elementos de conteúdo que a frase a ser gerada deve expressar. Esses tipos de relações resultam de regularidades semânticas gerais observadas entres formas lingüísticas (em geral em mais de uma língua). Portanto, elas são essentialmente independentes do domínio da aplicação, e constituem a interface entre (1) o processamento puramente *conceitual* do determinador do conteúdo e dos planejadores do discurso e das frases, e (2) o processamento puramente *lingüístico* do lexicalizador e do realizador sintático.

Existem dois tipos de sub-árvores semânticas:

- As de natureza *enciclopédica*, que consistem em conceitos na sua totalidade provenientes do modelo do domínio, vistos de uma perspectiva ontológica particular definida pelo planejador de frase.

[12] No caso das palavras invariáveis.

- As de natureza *retórica*, que consistem em conjuntos de atributos montados pelo planejador de frases a partir de vários conceitos do modelo do domínio.

A Fig. 19 apresenta dois exemplos de árvore semântica, L_3^e e L_4^e, para dois grupos de frases sinônimas. A saída do lexicalizador L_4^s correspondente ao exemplo à direita da Fig. 19 é dada na Fig. 20.

5.3.2 Sub-tarefas do Lexicalizador

Para mapear uma árvore semântica em uma árvore temática lexicalizada, o lexicalizador precisa escolher, para cada constituinte da árvore temática:

- Suas palavras de classes abertas, que devem expressar o conceito correspondente (*e.g.*, *"victory"* para `beat`, e *"home"* para `host` em L_4^s da Fig. 20);
- Sua função gramatical (*e.g.*, em L_4^s da Fig. 20, `classificador` e `qualificador`, respectivamente, para os papéis temáticos `locação` e `oposição` de P_1^s da Fig. 19);
- Sua categoria sintática (*e.g.*, em L_4^s da Fig. 20, `PP` para o papel `locação` de P_1^s da Fig. 19);
- Seus traços sintáticos não-default (*e.g.*, em L_4^s da Fig. 20 `[nome-proprio sim]`, visto que o default para a palavra-raiz do um SN é um substantivo comum).

5.3.3 Fatores que Influenciam na Lexicalização

A tarefa da lexicalização é uma das mais complexas, devido à grande variedade de fatores que exercem influência sobre ela, além do conteúdo a expressar (*i.e.*, os conceitos da árvore semântica de entrada). Encontramos:

- Fatores *não lingüísticos* como:
  - O contexto *enciclopédico, i.e.,* as relações desses conceitos com o resto do modelo do domínio;
  - O contexto *discursivo, i.e.,* o conteúdo e a forma das frases precedentes no texto gerado;
  - O contexto *interpessoal, i.e.,* os objetivos, planos e posição social do leitor;
  - O conhecimento do domínio e o vocabulário assumidos do leitor;
  - O contexto áudio-visual da apresentação nos geradores de linguagem natural integrados a sistemas multimídia;
- Fatores *lingüísticos* como:
  - As restrições de co-ocorrência lexical entre palavras;
  - As propriedades sintáticas particulares de cada palavra.

A seguir, veremos um exemplo de cada tipo de restrição:

- *Influência do contexto enciclopédico*: considere a escolha das palavras para se referir a um time de esporte em um artigo de jornal. Se um time é o único de sua cidade na competição descrita, o nome dessa cidade pode ser ecolhido (*e.g., "Chicago"* para os Chicago Bulls no NBA[13]). No caso de haver mais de um time da mesma cidade, o nome do clube deve ser escolhido, para evitar ambigüidade (*e.g., "the Lakers"* para os Los Angeles Lakers, para evitar confusão com os Los Angeles Clippers).

[13] National Basketball Association, o campeonato profissional norte-americano de basquetebol.

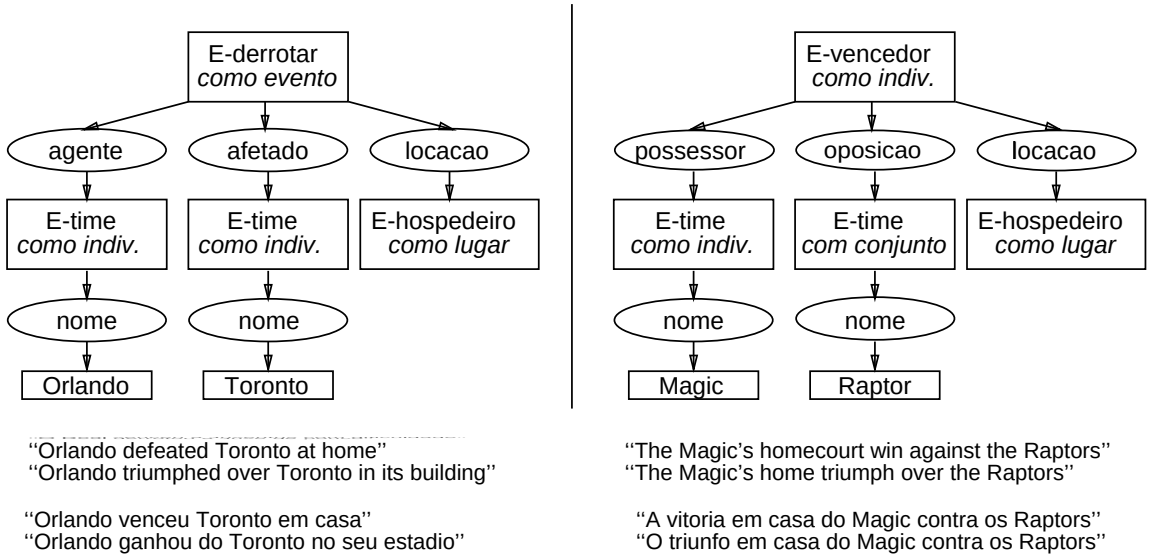

Figura 19: Exemplos de entrada do lexicalizador: L_3^e na esquerda e L_4^e na direita.

Saída do lexicalizador L_4^s:

$$\left[\begin{array}{ll} categoria & SN \\ palavra-raiz & \text{“}triumph\text{''} \\ possessor & \left[\begin{array}{ll} categoria & SN \\ palavra-raiz & \text{“}Magic\text{''} \\ nome-proprio & sim \end{array}\right] \\ classificador & \left[\begin{array}{ll} categoria & Subs \\ palavra-raiz & \text{“}home\text{''} \end{array}\right] \\ qualificador & \left[\begin{array}{lll} categoria & & SP \\ palavra-raiz & & \text{“}over\text{''} \\ complemento & \left[\begin{array}{ll} categoria & SN \\ palavra-raiz & \text{“}Raptor\text{''} \\ nome-proprio & sim \end{array}\right] & \end{array}\right] \end{array}\right]$$

Saída do gerador: F_4: *"The Magic's home triumph over the Raptors" (A vitoria do Magic contra os Raptors).*

Figura 20: Exemplo de saída do lexicalizador a partir da entrada L_i^e.

- *Influência do contexto discursivo*: considere a escolha das palavras para uma segunda referência, abreviadas para o indicador financeiro inicialmente introduzido no discurso pelas palavras *"The Dow Jones average"*. O gerador ANA, introduzido na seção 5.1.1, escolhe as palavras *"The Dow Jones"*, e jamais *"The average"*, para evitar ambigüidade com qualquer outro indicador financeiro que, como o *"Dow"*, são medidas de um conjunto de valores individuais.
- *Influência do contexto interpessoal*: considere a escolha das palavras *"combatentes da liberdade" vs. "terroristas"*, para referir-se a um grupo de guerrilheiros, a depender da orientação política desejada.
- *Influência do conhecimento do domínio assumido do leitor*: considere a escolha de palavras de conhecimento geral, como *"grande tubarrão branco"*, em contraste com palavras técnicas, como *"charcharodon charcharias"*, para referir-se à mesma espécie de animais.
- *Influência do contexto da apresentação em um gerador multimídia*: considere o caso uma apresentação que coordena texto e gráficos para explicar os modos de operação de um aparelho complexo, e em particular de uma referência ao botão de tensão no painel de controle. Ao inves de usar uma referência puramente lingüística (*e.g, "o botão de tensão"*) – que pode ser ineficiente se o painel tem muitos botões com rótulos pequenos – pode-se fazer uma referência coordenada usando ambos os mídia (*e.g. "o botão indicado da seta verde"*. Para gerar esse tipo de referência coordenada, o lexicalizador precisa ter acesso aos meta-objetos introduzidos para esse efeito pelo gerador de gráficos (neste caso, a seta verde).

Todos os fatores exemplificados até este ponto provêm de outras fontes de conhecimento que o léxico (entrada, modelo do domínio, do usuário e do discurso). Uma outra influência central sobre a lexicalização é *interno* ao léxico: restrições de co-ocorrência entre palavras, independentemente da sua semântica ou sintaxe. Como exemplo desses fatores *interlexicais*, considere os pares < verbo , substantivo-objeto > em expressões em inglês para igualar ou melhorar recordes. Nesse contexto, os verbos *"to post"* e *"to break"* são sinônimos, assim como os substantivos *"record"* e *"high"*. No entanto, restrições de co-ocorrência lexical proíbem dois dos quatro pares de combinações entre essas quatro palavras:

- *"to post a high"*
- $\star$ *"to post a record"*
- *"to break a record"*
- $\star$ *"to break a high"*

Como exemplo da influência das propriedades sintáticas especiais das palavras, considere verbos que não podem ser usados na voz passiva. Por exemplo, em inglês, *"to have"* pode ser usado em lugar de *"to score"* em frases ativas:

- *"He scored 29 points"*
- *"He had 29 points"*

A vantagem de usar "to have" é que ele é mas versátil semanticamente, e então permite formas concisas com conjunção embutida, que *"to score"* não permite:

- *"He had 29 points and 12 rebounds"*
- $\star$ *"He scored 29 points and 12 rebounds"*
- *"He scored 29 points and grabbed 12 rebounds"*

No entanto, se a voz passiva deve ser usada (por exemplo, para colocar o foco sobre o valor da estatística, em lugar de sobre o autor da ação), o verbo *"to have"* não pode ser usado (e conseqüentemente, conjunção embutida também não):

- *"29 points were scored by him"*
- $\star$ *"29 points were had by him"*

Esse último exemplo ilustra como fatores de natureza diversa (semânticos e sintáticos, neste caso) interagem de maneira sutil para restringir a escolha de uma palavra para expressar um conceito em um dado contexto de geração.

5.3.4 Granularidade do Léxico

Podemos distinguir dois tipos principais de léxicos computacionais para a geração de texto:

- Os léxicos *frasais*, cujas entradas mapeiam, simultaneamente, mais de dois conceitos em padrões de frases;
- Os léxicos *de palavras isoladas*, cujas entradas apenas mapeiam um ou dois conceitos em uma única palavra, ou em locuções contendo uma única palavra de classe aberta.

Para comparar essas duas abordagens, consideremos por exemplo a frase complexa a seguir: F_5: *"Dallas, TX – Charles Barkley tied a season high with 42 points and Danny Ainge came off the bench to add 21 Friday night as the Phoenix Suns handed the Dallas Mavericks their league worst 13th defeat in a row at home 123 - 97."*

Gerar F_5 com um léxico frasal consistiria apenas em combinar os três padrões de frases seguintes:

- "X tie a season high with N point"
- "Y come off the bench to add M point"
- "T hand Z its/their league worst Ith defeat in a row at home P-Q"

Em constraste, com um léxico de palavras individuais, gerar F_5 necessitaria combinar as 12 entradas seguintes:

- "tie", "season", "high", "point"
- "come off the bench", "add"
- "hand", "league", "worst", "defeat", "in a row", "home"

A partir deste exemplo, fica claro que um gerador usando um léxico frasal compõe textos *dinamicamente* porém a partir de (padrões de) frases *pré-fabricadas*. Portanto, esse tipo de gerador fica, em termos de composicionalidade, classificado entre um gerador que usa um léxico de palavras isoladas e um sistema que usa textos pré-fabricados.

Essa abordagem evita o desenvolvimento de mecanismos sofisticados necessários para controlar simultaneamente os fatores não lingüístico e lingüísticos sobre a lexicalização (enumerados na seção precedente). Os fatores lingüísticos, restrições de co-ocorrência lexical e propriedades especiais das palavras, não precisam ser considerados durante a geração de um texto. Eles só são considerados durante a composição do léxico frasal, a fim de assegurar que todas as suas entradas satisfazem essas restrições.

No entanto, o preço a pagar para essa simplificação é alto: redução dramática da extensibilidade do léxico. Considere, por exemplo, aumentar um léxico de palavras isoladas com um sinônimo e um antônimo para cada uma das 12 entradas necessárias para gerar F_5. Com apenas essas 24 novas entradas, o gerador poderá gerar aproximadamente[14] 1,594,232 ($3^4 \times 3^3 \times 3^6$) frases de estrutura semelhante à de F_5. Cobrir a mesma sub-linguagem com um léxico frasal requereria que este fosse aumentado com 837 (81 + 27 + 729) novas entradas.

5.3.5 Paradigmas Computacionais Usados para a lexicalização

A lexicalização é, muito provavelmente, a sub-tarefa da geração de linguagem natural para a qual foi usada a maior variedade de formalismos computacionais, incluindo:

- Redes de discriminação [Goldman, 1975] [Danlos, 1986].
- Regras de produção [Kukich, 1988].
- Classificação de conceitos [Reiter, 1991].
- Cobertura de grafos conceituais [Nogier & Zock, 1991].
- Unificação funcional [Robin, 1994] [Elhadad *et al.*, 1997].

5.4 Assuntos Avançados

Os principais assuntos avançados em geração de linguagem natural que constituem o foco da pesquisa atual na área são:

- **Modelagem do usuário**, tanto o uso de modelos estáticos para auxiliar na determinação do conteúdo e na organização do discurso [Paris, 1993], ou na organização da frase e na lexicalização [McKeown *et al.*, 1993] [Elhadad, 1993b], como a manutenção de modelos em domínios dinâmicos [Dale, 1992].
- **Estilística**, a tentativa de definir e usar um conjunto de regras de estilo com um nível de precisão necessário para controlar a organização do discurso, a organização das frases e a lexicalização em um gerador [Hovy, 1988] [BenHassine *et al.*, 1991] [Van der Linden & Martin, 1995].
- **Saída multilíngüe**, como nos geradores bilíngüe inglês-francês (desenvolvidos no Canadá) de boletins meteorológicos [Bourbeau *et al.*, 1990], de resumos de *logs* de monitoria de sistemas operacionais [Carcagno & Iordanskaja, 1993] e de estatísticas de desemprego [Iordanskaja *et al.*, 1994]; cada um desses geradores contém componentes de determinação do conteúdo e de organização do discurso *únicos*, independente da língua de saída; porém necessitam de dois componentes de organização das frases, de lexicalização e de realização sintática, um para cada língua.
- **Saída hipertexto**, usada principalmente em documentação automática [Reiter *et al.*, 1992] [Johnson, 1994] e para levantar questões sobre a organização de um discurso *não-linear*, com todas suas conseqüências intrigantes sobre o uso da anáfora e da elipse.
- **Saída falada e coordenação com outras mídia**, em particular a coordenação temporal entre uma saída falada, comentando uma série de gráficos [Dalal *et al.*, 1996]; essa dimensão de "tempo real" da geração de linguagem natural foi também estudada no contexto unimídia [Rubinoff, 1992].

[14] *i.e.*, ignorando possíveis restrições de co-ocorrência entre essas novas entradas, que poderiam diminuir um pouco esse número, contudo sem afetar sua ordem de grandeza.

- **Portabilidade e re-utilização de componentes**, o desenvolvimento de componentes re-utilizáveis em domínios de aplicação diferentes, como os realizadores sintáticos SURGE [Elhadad & Robin, 1996], NIGEL [Mann & Matthiessen, 1983] e MUMBLE-86 [Meteer *et al.*, 1987], a ontologia de senso comum UPPER-MODEL [Bateman *et al.*, 1990] e as regras de revisão de STREAK [Robin, 1996a].
- **Aquisição de conhecimento e corpora textuais**, o desenvolvimento de ferramentas para a aquisição semi-automática de estruturas de conhecimento utilizá veis por um gerador, a partir de análises estatísticas de corpora textuais; por exemplo, o sistema XTRACT [Smadja & McKeown, 1990], que automaticamente adquiriu as restrições de co-ocorrências lexicais implementadas no gerador COOK [Smadja & McKeown, 1991], ou a ferramenta CREP [Duford, 1993], que foi usada para semi-automaticamente adquirir as regras de revisão implementadas no gerador STREAK [Robin, 1994].
- **Gramáticas e léxicos reversíveis** [Strzalkowski, 1994], um esforço para desenvolver gramáticas e léxicos que possam ser utilizados tanto para a interpretação como para a geração, a fim de evitar a duplicação do mesmo conhecimento nas aplicações que requerem as duas direções de processamento (como as interfaces de banco de dados ou a tradução automática); no estado atual, apesar de alguns *formalismos* reversíveis promissores terem sido propostos (como as gramáticas de adjunção de árvores sincronizadas [Shieber & Shabes, 1991]), todas as gramáticas e léxicos computacionais de cobertura abrangente existentes ainda ficam codificados em formalismos especializados (para a interpretação ou para a geração).
- **Arquiteturas alternativas**, por exemplo, nas quais a decomposição do gerador em componentes *não* segue a decomposição em tarefas definidas na seção 5.1.4 [Danlos, 1986], ou o fluxo de controle entre componentes é bidirecional [Appelt, 1985] [Hovy, 1988] [McKeown *et al.*, 1993] [Rubinoff, 1992].
- **Revisão e geração incremental**, um outro tipo de arquitetura alternativa na qual apenas uma parte do texto é inicialmente gerado, e depois incrementalmente revisado para gradualmente incluir elementos de conteúdo complementares no contexto da parte já gerada; esse modelo foi desenvolvido para a modelagem cognitiva de geração de frases por pessoas [De Smedt, 1990], para respeitar regras estilísticas e para a geração de frases muito complexas [Robin & McKeown, 1993].
- **Robustez e modelos estáticos da língua**, o desenvolvimento de geradores que conseguem gerar saídas parcialmente corretas[15] a partir de entradas, ou usando fontes de conhecimento incompletas; por exemplo, o gerador do tradutor automático JAPANGLOSS [Knight & Hatzivassiloglou, 1995], onde a maioria das saídas incorretas causadas pela ausência de alguma entrada lexical ou regra gramatical são filtradas por um modelo estatístico do inglês, baseado na probabilidade de cada palavra seguir qualquer par de palavras, compilada sobre grande corpora de textos em inglês.
- **Avaliação empírica**, a definição de métodos para avaliar *quantitativamente e objetivamente* várias caraterísticas de geradores de linguagem natural: como a cobertura (semântica, lexical ou sintática) [Kukich, 1983], ou precisão (semântica ou estilística) da implementenção [Lester, 1994] [Van der Linden & Martin, 1995], a robustez e extensibilidade do modelo de geração subjacente [Robin, 1996b], ou a portabilidade das estruturas de conhecimentos usadas [Robin, 1996a].

[15] Em lugar de não gerar nenhuma saída, ou de gerar saídas totalmente erradas, como na maiora dos geradores atuais.

Referências

[Allen, 1983] Allen, J.F. Recognizing intentions from natural language utterances. In Brady, M. & Berwick, R.C , editors, *Computational Models of Discourse*, pages 107–166. MIT Press, 1983.

[Allen, 1995] Allen, J.F. *Natural Language Understanding.* Benjamin/Cummings, 2nd edition, 1995.

[Appelt, 1985] Appelt, D. *Planning Natural Language Utterances.* Studies in Natural Language Processing. Cambridge University Press, 1985.

[Bateman *et al.*, 1990] Bateman, J.A., Kasper, R.T., Moore, J.D., & Whitney, R.A. A general organization of knowledge for natural language processing: the penman upper-model. Technical report, ISI, Marina del Rey, CA, 1990.

[BenHassine *et al.*, 1991] BenHassine, N., DiMarco, C., & Randall, N. Controlling style in natural language generation. In *Proceedings of the IJCAI-91 Workshop on Decision Making Throughout the Generation Process*, pages 18–25, Sydney, Australia, 1991.

[Bourbeau *et al.*, 1990] Bourbeau, L., Carcagno, D., Goldberg, E., Kittredge, R., & Polguère, A. Bilingual generation of weather forecasts in an operations environment. In *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki University, Finland, 1990. COLING.

[Carcagno & Iordanskaja, 1993] Carcagno, D. & Iordanskaja, L. Content determination and text structuring: two interrelated processes. In Horacek, H. & Zock, M., editors, *New Concepts in Natural Language Generation: Planning, Realization and Systems.* Frances Pinter, London and New York, 1993.

[Chomsky, 1956] Chomsky, N. Three models for the description of language. *IRE Transactions PGIT*, 2:113–124, 1956.

[Chomsky, 1973] Chomsky, N. Conditions on transformations. In Anderson, S.R. & Kiparsky, P., editors, *A Festschrift for Morris Halle.* Holt, Rinehart and Winston, Inc, 1973.

[Crystal, 1987] Crystal, D. *The Cambridge Encyclopedia of Language.* Cambridge University Press, 1987.

[Crystal, 1991] Crystal, D. *A Dictionary of Linguistics and Phonetics.* Basil Blackwell, Oxford, UK., 3rd edition, 1991.

[Dalal *et al.*, 1996] Dalal, M., Feiner, S.K., McKeown, K.R., Jordan, D., Allen, B., & alSafadi, Y. Magic: An experimental system for generating multimedia briefings about post-bypass patient status. In *Proceedings of the American Medical Informatics Association*, Washington, D.C., October 1996.

[Dale & Delin, 1991] Dale, R. & Delin, J. Theories of discourse structure. In *Lecture Notes for the Third European Summer School in Language, Logic & Information.* University of Saarlandes, Saarbrucken, Germany, 1991.

[Dale, 1992] Dale, R. *Generating Referring Expressions.* ACL-MIT Press Series in Natural Language Processing, Cambridge, Ma., 1992.

[Danlos, 1986] Danlos, L. *The linguistic basis of text generation.* Studies in Natural Language Processing. Cambridge University Press, 1986.

[De Smedt, 1990] De Smedt, K.J.M.J. *Incremental sentence generation: a computer model of grammatical encoding.* Samson-Sijthoff, The Netherlands, 1990.

[Dubois *et al.*, 1973] Dubois, J., Giacomo, M., Guespin, L., Marcellesi, C., Marcellesi, J., & Mevel, J. *Dicionário de Linguistica.* Editora Cultrix, São Paulo, 1973.

[Duford, 1993] Duford, D. Crep: a regular expression-matching textual corpus tool. Technical Report CUCS-005-93, Columbia University, 1993.

[Elhadad & Robin, 1996] Elhadad, M. & Robin, J. An overview of SURGE: a re-usable comprehensive syntactic realization component. Technical Report 96-03, Ben Gurion University of the Negev, Mathematics and Computer Science Department, Beer Sheva, Israel, 1996.

[Elhadad *et al.*, 1997] Elhadad, M., McKeown, K., & Robin, J. Floatings constraints in lexical choice. *Computational Linguistics*, 23(2):195-239, 1997.

[Elhadad, 1993a] Elhadad, M. Fuf: The universal unifier - user manual, version 5.2. Technical Report CUCS-038-91, Columbia University, 1993.

[Elhadad, 1993b] Elhadad, M. *Using argumentation to control lexical choice: a unification-based implementation.* PhD thesis, Computer Science Department, Columbia University, 1993.

[Fawcett, 1987] Fawcett, R.P. The semantics of clause and verb for relational processes in english. In Halliday, M.A.K. & Fawcett, R.P., editors, *New developments in systemic linguistics.* Frances Pinter, London and New York, 1987.

[Fillmore, 1968] Fillmore, C.J. The case for case. In Bach, E. & Harms, R., editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart and Winston, 1968.

[Fromkin & Rodman, 1988] Fromkin, V. & Rodman, R. *An Introduction to Language.* Holt, Rinehart and Winston, Inc., 4th edition, 1988.

[Gazdar & Mellish, 1989] Gazdar, G. & Mellish, C. *Natural Language Processing in Prolog: An Introduction to Computational Linguistics.* Addison-Wesley Publishing Company, 1989.

[Gazdar *et al.*, 1985] Gazdar, G., Klein, E., Pullum, G., & Sag, I. *Generalised Phrase Structure Grammar.* Basil Blackwell, 1985.

[Goldman, 1975] Goldman, N. Conceptual generation. In Schank, Roger , editor, *Conceptual Information Processing*, pages 289–374. North-Holland, Amsterdam, 1975.

[Grice, 1969] Grice, H.P. Utterer's meaning and intention. *Philosophical Review*, 68(2):147–177, 1969.

[Grosz & Sidner, 1986] Grosz, B.J. & Sidner, C.L. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[Grosz *et al.*, 1983] Grosz, B.J., Joshi, A.K., & Weinstein, S. Providing a unified account of definite noun phrase in discourse. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 44–50. ACL, 1983.

[Grosz, 1977] Grosz, B.J. The representation and use of focus in a system for understanding dialogs. In *Proceedings of the Fifth Int. Joint Conference on AI*, pages 67–76. IJCAI'77, 1977.

[Halliday, 1994] Halliday, M.A.K. *An introduction to functional grammar.* Edward Arnold, London, 1994. 2nd Edition.

[Hobbs, 1978] Hobbs, J.R. Resolving pronoun reference. *Lingua*, 44:311–338, 1978.

[Hobbs, 1979] Hobbs, J.R. Coherence and coreference. *Cognitive Science*, 3(1):67–90, 1979.

[Hovy, 1988] Hovy, E. *Generating natural language under pragmatic constraints.* L. Erlbaum Associates, Hillsdale, N.J., 1988.

[Hovy, 1991] Hovy, E. Approaches to the planning of coherent text. In Paris, C., Swartout, W., & W.C., Mann., editors, *Natural language generation in artificial intelligence and computational linguistics.* Kluwer Academic Publishers, Boston, 1991.

[Iordanskaja *et al.*, 1994] Iordanskaja, L., Kim, M., Kittredge, R., Lavoie, B., & Polguère, A. Generation of extended bilingual statistical reports. In *Proceedings of the 15th International Conference on Computational Linguistics.* COLING, 1994.

[Jackendoff, 1990] Jackendoff, R. *Semantic Structures.* MIT Press, Cambridge, MA, 1990.

[Johnson, 1994] Johnson, W.L. Dynamic (re)generation of software documentation. In *Proceedings of the 4th Systems Reengineering Technology Workshop*, pages 57–66, Johns Hopkins University, 1994.

[Kamp & Reyle, 1993] Kamp, H. & Reyle, U. *From Discourse to Logics.* Kluwer Academic Publishers, Dordrecht, Netherlands, 1993.

[Kay, 1979] Kay, M. Functional grammar. In *Proceedings of the 5th Annual Meeting of the Berkeley Linguistic Society*, 1979.

[Knight & Hatzivassiloglou, 1995] Knight, K. & Hatzivassiloglou, V. Two-levels, many-paths generation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 252–260, Boston, MA, 1995. ACL.

[Kukich *et al.*, 1994] Kukich, K., McKeown, K., Shaw, J., Robin, J., Morgan, N., & Phillips, J. User-needs analysis and design methodology for an automated document generator. In Zampolli, A., Calzolari, N., & Palmer, M., editors, *Current Issues in Computational Linguistics: In Honour of Don Walker.* Kluwer Academic Press, Boston, 1994.

[Kukich, 1983] Kukich, K. *Knowledge-based report generation: a knowledge engineering approach to natural language report generation.* PhD thesis, University of Pittsburgh, 1983.

[Kukich, 1988] Kukich, K. Fluency in natural language reports. In McDonald, D.D. & L., Bolc , editors, *Natural Language Generation Systems.* Springer-Verlag, New York, NY, 1988.

[Lester, 1994] Lester, J.C. *Generating natural language explanations from large-scale knowledge bases.* PhD thesis, Computer Science Department, Universtity of Texas at Austin, New York, NY, 1994.

[Mann & Matthiessen, 1983] Mann, W.C. & Matthiessen, C.M. Nigel: a systemic grammar for text generation. Technical Report ISI/RR-83-105, ISI, Marina del Rey, CA, 1983.

[Mann & Thompson, 1987] Mann, W.C. & Thompson, S. Rhetorical structure theory: description and constructions of text structures. In Kempen, Gerard , editor, *Natural Language Generation: New Results in Artificial Intellligence, Psychology and Linguistics*, pages 85–96. Martinus Ninjhoff Publishers, 1987.

[McKeown & Radev, 1995] McKeown, K. & Radev, D. Generating summaries of multiple news articles. In *Proceedings of SIGIR*, Seattle, Wash., July 1995.

[McKeown & Swartout, 1987] McKeown, K. & Swartout, W.R. Language generation and explanation. *The Annual Review of Computer Science*, (2):401–449, 1987.

[McKeown *et al.*, 1990] McKeown, K. R., Elhadad, M., Fukumoto, Y., Lim, J.G., Lombardi, C., Robin, J., & Smadja, F.A. Text generation in comet. In Dale, R., Mellish, C.S., & Zock, M., editors, *Current Research in Natural Language Generation*, pages 103–140. Academic Press, 1990.

[McKeown *et al.*, 1993] McKeown, K., Robin, J., & Tanenblatt, M. Tailoring lexical choice to the user's vocabulary in multimedia explanation generation. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics.* ACL, 1993.

[McKeown, 1985] McKeown, K. *Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Studies in Natural Language Processing. Cambridge University Press, 1985.

[Meteer *et al.*, 1987] Meteer, M.W., McDonald, D.D., Anderson, S.D., Forster, D., Gay, L.S., Huettner, A.K., & Sibun, P. Mumble-86: Design and implementation. Technical Report COINS 87-87, University of Massachussets at Amherst, Ahmerst, Ma., 1987.

[Nogier & Zock, 1991] Nogier, J.F. & Zock, M. Lexical choice as pattern matching. In Nagle, Nagle & Eklund, , editors, *Current Directions in Conceptual Structures Research.* 1991.

[Paris, 1993] Paris, C.L. *User Modelling in Text Generation.* Francis Pinter Publishers, London, 1993.

[Passoneau *et al.*, 1996] Passoneau, R., Kukich, K., Robin, J., Hatzivassiloglou, V., Lefkowitz, L., & Jing, H. Generating summaries of workflow diagrams. In *Proceedings of the International Conference on Natural Language Processing and Industrial Applications (NLP-IA'96)*, Moncton, New Brunswick, Canada, 1996.

[Quirk *et al.*, 1985] Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. *A comprehensive grammar of the English language*. Longman, 1985.

[Reichman-Adar, 1984] Reichman-Adar, R. Extended person-machine interface. *Artificial Intelligence*, 22:157–218, 1984.

[Reiter *et al.*, 1992] Reiter, E., Mellish, C., & Levine, J. Automatic generation of on-line documentation in the idas project. In *Proceedings of the ACL Applied Natural Language Conference*, Trento, Italy, April 1992.

[Reiter, 1991] Reiter, E.B. A new model for lexical choice for open-class words. *Computational Intelligence*, (7), December 1991.

[Rich & Knight, 1994] Rich, E. & Knight, K. *Inteligência Artificial*. Makron Books, 1994.

[Robin & McKeown, 1993] Robin, J. & McKeown, K. Corpus analysis for revision-based generation of complex sentences. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 365–372. AAAI, 1993.

[Robin & McKeown, 1996] Robin, J. & McKeown, K. Empirically designing and evaluating a new revision-based model for summary generation. *Artificial Intelligence*, 85, August 1996. Special Issue on Empirical Methods.

[Robin, 1994] Robin, J. Automatic generation and revision of natural language summaries providing historical background. In *Proceedings of the 11th Brazilian Symposium on Artificial Intelligence*, Fortaleza, CE, October 1994.

[Robin, 1996a] Robin, J. Evaluating the portability of revision rules for incremental summary generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, California, 1996. ACL.

[Robin, 1996b] Robin, J. Evaluating the robustness and scalability of revision-based natural language generation. In *Proceedings of the 13th Brazilian Symposium on Artificial Intelligence*, Curitiba, PN, October 1996.

[Rubinoff, 1992] Rubinoff, R. *Negotiation, feedback and perspective within natural language generation*. PhD thesis, Computer Science Department, University of Pennsylvania, 1992.

[Searle, 1971] Searle, J.R. What is a speech acts. In Searle, J.R., editor, *Oxford Readings in Philosophy*, pages 39–53. Oxford University Press, London, 1971.

[Shieber & Shabes, 1991] Shieber, S.M. & Shabes, Y. Generation and synchronous tree-adjoining grammars. *Computational Intelligence*, 7(4):220–228, December 1991.

[Shieber, 1984] Shieber, S.M. The design of a computer language for linguistic information. In *Proceedings of the 10th International Conference on Computational Linguistics*, pages 362–366, Stanford University, CA, 1984. COLING.

[Sidner, 1983] Sidner, C.L. Focusing in the comprehension of definite anaphora. In Brady, M. & Berwick, R.C , editors, *Computational Models of Discourse*, pages 267–330. MIT Press, 1983.

[Smadja & McKeown, 1990] Smadja, F. & McKeown, K. Automatically extracting and representing collocations for language generation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 252–9, Pittsburgh, Pa., June 1990.

[Smadja & McKeown, 1991] Smadja, F.A. & McKeown, K. Using collocations for language generation. *Computational Intelligence*, 7(4):229–239, December 1991.

[Strzalkowski, 1994] Strzalkowski, T. (Ed.) . *Reversible grammars in natural language processing*. Kluwer Academic Publishers, Boston, 1994.

[Talmy, 1985] Talmy, L. Lexicalization patterns: semantic structure in lexical form. In Shopen, T., editor, *Grammatical categories and the lexicon*, volume 3 of *Language typology and syntactic description*. Cambridge University Press, 1985.

[Van der Linden & Martin, 1995] Van der Linden, K. & Martin, J.H. Expressing rhetorical relations in instructional texts: a case study of the purpose relation. *Computational Linguistics*, 21(1):29–58, 1995.

[Webber, 1981] Webber, B.L. Discourse model synthesis: Preliminaries to reference. In Joshia, A.K., Webber, B.L., & Sag, I., editors, *Elements of Discourse Understanding*, pages 283–299. Cambridge University Press, 1981.

[Winograd, 1983] Winograd, T. *Language as a cognitive process.* Addison-Wesley, 1983.

[Winston, 1992] Winston, P.H. *Artificial Intelligence.* Addison-Wesley, 1992.

[Woods, 1970] Woods, W.A. Transition network grammars for natural language analysis. *Communications of ACM*, 13(10):591–606, 1970.