

Tecnologia Adaptativa Aplicada ao Processamento Computacional de Língua Natural

R. L. A. Rocha

Resumo—Este artigo descreve algumas técnicas adaptativas aplicadas ao processamento computacional de línguas naturais. Começa com uma apresentação de caráter histórico e então descreve como a tecnologia adaptativa pode ser usada para lidar com o processamento computacional de línguas naturais. O artigo é finalizado mostrando algumas das técnicas utilizadas pelos pesquisadores do grupo de pesquisa em tecnologia adaptativa do Laboratório de Linguagens e Tecnologia Adaptativa.

Palavras Chave—Linguagens Formais, Teoria de Autômatos, Tecnologia Adaptativa, Processamento de Linguagem Natural

I. INTRODUÇÃO

O PROCESSAMENTO computacional de língua natural é uma área extremamente complexa, que tem sido objeto de pesquisa há várias décadas [4], [15]. Complexa porque envolve processar a emissão (síntese) de texto/som, recepção de texto/som e compreensão de texto. Alguns dos diversos aspectos envolvidos que tornam o processamento de língua natural, além de complexo, computacionalmente mais difícil são:

- Compreensão da linguagem recebida como entrada para um sistema, seja ele computacional ou humano, dentro de regras dependentes de contexto:
 - Entonação da frase (por exemplo, interrogativa, exclamativa, etc.);
 - Interlocutor do diálogo, palavras técnicas (dependentes de contexto), etc.;
 - Tema descrito no texto, etc.;
- A representação da língua usada (falada e/ou escrita) por algum segmento populacional como característica central, sendo, portanto, dinâmica:
 - Se a saída computacional se der por meio sonoro, o regionalismo imprime uma dificuldade ainda mais acentuada.;
- Conexão com áreas distintas, tais como: Computação, Lingüística, Psicologia, Fonética,

Processamento digital de sinais.

O tratamento das línguas naturais usualmente é associado a métodos usados em Inteligência Artificial, como, por exemplo, métodos de base estatística ou métodos descritivos baseados em gramáticas (chamados de racionais). Um dos problemas centrais é a necessidade de elevado desempenho computacional, devido às diversas possibilidades de interpretação de uma frase e da necessidade de respostas em tempo exequível.

Os principais problemas enfrentados no processamento de língua natural referem-se ao tratamento das ambigüidades e dos não-determinismos:

- Ambigüidade léxica: elementos com as mesmas categorias lexicais, mas com formas diferentes (Ex.: foi – verbo ir ou verbo ser); elementos com a mesma forma básica, porém com categorias diferentes (Ex.: complementar – verbo ou adjetivo);
- Ambigüidade flexiva: elementos com as mesmas formas básicas, mas diferentes categorias (Ex.: amamos – verbo no presente ou no passado);
- Ambigüidade léxico-flexiva: elementos com formas básicas diferentes e categorias diferentes (Ex.: busca – verbo, substantivo);
- Ambigüidade sintática:
 - Na identidade do constituinte (Ex.: o homem com a bicicleta da China);
 - Coesão (Ex.: anáfora: amava sua irmã);
 - Coordenação (Ex.: homens e mulheres no Brasil);
- Ambigüidade semântica:
 - Léxica (Ex.: fato – acontecimento, ou terno);
 - Escopo (Ex.: não compre três garrafas de vinho, compre licor);
 - Papel temático (Ex.: o sacrifício da moça);
- Não-determinismos: diferentes alternativas de reconhecimento que podem se refletir em diferentes grafos de análise sintática.

A. Histórico e Conceitos Iniciais

O processamento de língua natural envolvendo

compreensão da linguagem requer informação léxica, sintática e também semântica, como foi rapidamente percebido desde o seu início. Abstraindo a questão da recepção de sinais e transformação em texto, cada componente (de texto) recebido deve ser fracionado para que a sintaxe da mensagem seja corretamente avaliada. Assim, divide-se o texto em componentes léxicos e avalia-se o papel sintático desses componentes. De acordo com o papel sintático, pode-se inferir mais adequadamente qual a semântica do texto recebido.

Inicialmente, os experimentos usavam intenso processamento simbólico, em alguma linguagem específica como LISP ou Prolog. Além disso, reduzia-se o foco a algum domínio restrito, como, por exemplo, relatórios financeiros, entrada de dados de produção.

Até os anos 1980, a pesquisa em PLN usava basicamente processamento simbólico por meio de gramáticas e bases de conhecimento. Por conta das dificuldades encontradas à época (não apenas tecnológicas, mas também conceituais – extrapolação de domínios restritos, por exemplo), iniciou-se uma mudança de proposta para PLN, baseada em dados empíricos (*corpora* e treinamento). O foco passou a ser a utilização de métodos estatísticos.

1) O uso de Técnicas Estatísticas

Os primeiros resultados animadores apareceram no processamento de sinais (som), através do uso de modelos de Markov (HMM).

A exploração de modelos estatísticos nas outras etapas da PLN passa a ser empreendida e surgem os primeiros etiquetadores morfológicos. A seguir, esses etiquetadores passam a utilizar uma estratégia mista, não puramente estatística, obtendo resultados de acerto > 95%.

Com relação ao analisador gramatical, a aposta passou a ser no uso de gramáticas livres de contexto probabilísticas (PCFG). Um dos pioneiros no estudo e propositor de um modelo racionalista (em Prolog) é um dos incentivadores de PCFG.

Outra proposta é utilizar um modelo de n-palavras em sequência (bigramas, trigramas, etc.).

2) Propostas Atuais

Entre as propostas atuais, tem-se o uso de modelos estatísticos baseados em métodos racionais, o uso de regras lógicas dentro de modelos estatísticos, o uso de modelos racionais com base em algum método estatístico.

Utilizar algum modelo estatístico como base para escolha de parâmetros de modelos não estatísticos (diminuindo desta forma o não-determinismo e a ambigüidade) é também outra opção atualmente utilizada.

O uso indiscriminado de modelos estatísticos não resolveu os problemas mais graves de Processamento de Língua Natural, embora tenha revelado um aspecto importante, que é a necessidade de evitar resposta única.

Em termos lingüísticos, o estudo dos fenômenos da linguagem não busca meramente descrever os aspectos morfológico, léxico e sintático, mas, segundo Chomsky, estudar como a linguagem é aprendida (por crianças) e quais são as características comuns entre todas as linguagens.

Esses fenômenos não são aparentes e, portanto não são facilmente tratáveis via *corpus* de treinamento.

Outro aspecto levantado por Chomsky refere-se às características inatas (não aprendidas) das linguagens, o que tornaria inviável o uso de métodos estatísticos de aprendizado, se tais características fossem majoritariamente presentes.

Foi observado que há de fato um conjunto de características consideradas inatas (não aprendidas) nas linguagens naturais;

Chomsky propõe a existência de uma gramática universal.

a) A Questão da Gramática Universal

Estudos efetuados na área de lingüística e de psicologia (referindo-se à cognição) têm mostrado resultados extremamente interessantes no que tange à proposta de Chomsky. O primeiro resultado mostra estatisticamente que há uma coincidência muito forte entre as diversas gramáticas das linguagens naturais existentes. Este resultado por si só é um grande indicador de pesquisas na área de lingüística e computação, embora não prove a existência de uma gramática universal [16].

Entretanto, o processo de aquisição da linguagem por crianças é extremamente semelhante independentemente da linguagem. Essa característica levantada pelos estudos cognitivos praticamente elimina a possibilidade de que os efeitos culturais tenham papel central na definição da gramática básica [16]. Dessa maneira, o aprendizado da língua natural por crianças se dá mediante a existência de uma gramática inata e da mediação cultural (que refina a gramática).

Quando há diferenças acentuadas nas linguagens, o aprendizado de alguma característica de uma linguagem pode dar-se em momentos diferentes. Há coincidência na escolha da “gramática” pelas crianças, que alteram a “gramática” com o aprendizado (quando ocorre o efeito da cultura local).

Exemplo: Omissão de objetos e de sujeitos (pessoas) em frases – o aprendizado dessa característica na língua natural apresenta-se da mesma forma entre crianças chinesas e americanas [16].

Esses fenômenos não são aparentes e, portanto, não são facilmente tratáveis via *corpus* de treinamento. Outro aspecto levantado por Chomsky refere-se às características inatas (não aprendidas) das linguagens, o que tornaria inviável o uso de métodos estatísticos de aprendizado, se tais características fossem majoritariamente presentes.

Conforme observado, modelos híbridos podem obter resultados melhores do que modelos puramente estatísticos e, além disso, há evidências empíricas que suportam a existência de uma gramática universal. Essas considerações corroboram a busca por um modelo alternativo que possa atender à diversidade de aspectos enfrentados no processamento de língua natural e que, ao mesmo tempo, possa utilizar as características mais adequadas para tratar cada aspecto. Para obter esse resultado, a construção do modelo deve seguir um método que considere os demais modelos.

II. A QUESTÃO DO CONTEXTO – USO DE TÉCNICAS ADAPTATIVAS

Em uma linguagem natural, o problema da identificação e uso correto do contexto é particularmente complexo. A complexidade decorre do fato de os elementos de um discurso remeterem a partes distantes do próprio discurso (escrito ou falado), no qual um interlocutor é responsável por identificá-los nas partes.

Em qualquer conversa informal, os termos, nomes, eventos às vezes não são sequer explicitamente mencionados porque o discurso está inserido em um contexto. Assim, o conhecimento do interlocutor a respeito de elementos do discurso permite eliminar ambigüidades do discurso.

O contexto do discurso pode ser definido antecipadamente, *a priori*, por exemplo, participação de uma palestra sobre área específica.

O tratamento do contexto é complexo e a dependência de contexto não é uma característica exclusiva das linguagens naturais. As linguagens de programação apresentam essa característica, que é normalmente tratada pelos compiladores usando um modelo “semântico”, ou seja, usando técnicas de tratamento semântico para lidar com aspectos sintáticos dependentes de contexto [6].

A. O Problema do Contexto

Há características da estrutura das linguagens de programação que não podem ser descritas usando-se sintaxe livre de contexto, como por exemplo:

- Declaração de variáveis antes de seu uso.
- Declarações **end** <nome> em linguagens (como ADA) nas quais <nome> deve coincidir com o nome de algum subprograma.
- A compatibilidade de tipos poderia, em tese, ser feita, mas a complexidade requerida é muito grande [14].

Essas são características dependentes de contexto ou cujo tratamento é efetuado por meio de dependência de contexto (compatibilidade de tipos).

Portanto, a verificação de tais características sintáticas não pode ser feita por qualquer dispositivo livre de contexto. Esse conjunto de tarefas de verificação foi denominado semântica estática, embora nada tenha a ver com semântica. O motivo da denominação é o processo utilizado pelos compiladores para tratar essas questões.

Uma linguagem cujas construções precisem de informações presentes em partes distintas de suas cadeias é dita dependente de contexto.

Ex.: Seja L uma linguagem tal que $L = a^n b^n c^n$, o valor de n deve coincidir nas três subcadeias (prova-se usando o bombeamento para linguagens Livres de Contexto que esta linguagem não é Livre de Contexto [11]).

Portanto, as questões colocadas anteriormente quando aplicadas às linguagens de programação ilustram sua dependência de contexto.

Para as linguagens de programação, cujo processamento é necessariamente eficiente e de resposta rápida [1], torna-se

impraticável o uso de gramáticas ou de *parsers* dependentes de contexto para o seu tratamento.

O tratamento das questões relativas ao contexto levou a comunidade acadêmica a produzir algumas alternativas, das quais duas se destacaram [14]:

- Gramática de dois níveis: estabelecida por van Wijngaarden e utilizada na definição das características sintáticas da linguagem de programação Algol 60;
- Gramática de atributos: definida por Knuth e usada ainda hoje para tratamento da sintaxe dependente de contexto em compiladores.

B. Autômatos Adaptativos e Linguagens Dependentes de Contexto

O tratamento das questões relativas ao contexto por meio de gramáticas de dois níveis ou gramáticas de atributos apresenta duas proposições distintas:

- Definição formal da sintaxe dependente de contexto (gramáticas de dois níveis) e tratamento do contexto dentro do formalismo (gramáticas de dois níveis);
- Definição da sintaxe dependente de contexto por meio de elementos externos ao formalismo gramatical (gramáticas de atributos), forte vinculação com aplicação prática para construção de compiladores (gramáticas de atributos).

O uso de Tecnologia Adaptativa permite agregar características de ambas porque o modelo formal subjacente altera sua estrutura de modo a fazer frente às necessidades da linguagem; assim, pode-se obter:

- A definição formal da sintaxe dependente de contexto;
- O tratamento do contexto dentro do formalismo;
- A vinculação com aplicação prática.

1) O tratamento das Linguagens Usando Autômatos Adaptativos

A forma geral das transições do Autômato Adaptativo permite o tratamento das várias classes de linguagens na hierarquia de Chomsky, de forma progressiva [7].

As construções regulares são tratadas de forma simples (por meio de autômatos finitos), sem uso de pilha ou de ações adaptativas (por meio de autômatos de estados finitos).

As construções livres de contexto são tratadas através do uso da pilha presente no modelo (por meio do Autômato de Pilha Estruturado).

As construções dependentes de contexto são tratadas através das ações adaptativas, que permitem que o modelo de autômato altere sua topologia.

Em seu caso mais geral, o Autômato Adaptativo é capaz de lidar com construções recursivamente enumeráveis (como aceitadores), por conta de sua equivalência com a Máquina de Turing [6], [13].

Para as questões relativas ao contexto, a adaptatividade permite eliminar o uso de elementos externos ao modelo. Assim, o tratamento do contexto por meio de Autômatos

Adaptativos permite [6]:

- Eliminar pilhas explícitas;
- Eliminar tabelas de símbolos e palavras reservadas;
- Coletar nomes;
- Tratar sintaticamente escopos;
- Tratar sintaticamente os tipos;
- Tratar sintaticamente declarações e chamadas de funções;
- Verificar sintaticamente argumento e parâmetro;
- Tratar declaração e expansão de macros;
- Tratar linguagens extensíveis.

2) Exemplo: Tratamento Sintático de Tipos

A declaração de algum objeto em uma linguagem de programação para o seu uso posterior define atributos do objeto, que devem ser verificados quanto à coerência entre a declaração e o uso.

A verificação é feita através de ações adaptativas que alteram a topologia do autômato, para aceitar somente as construções sintáticas em que sejam obedecidos os vínculos corretos entre os tipos declarado e esperado.

Desenvolve-se abaixo um exemplo em que há dois tipos de variáveis e, assim, dois tipos de comandos de atribuição, envolvendo inteiros e reais.

Os comandos de atribuição compõem-se de duas partes, uma variável de destino e uma expressão aritmética (extremamente simplificada, com números e variáveis).

Encontrado o identificador que representa a variável destino da atribuição, utiliza-se o tipo a ele associado como referência a uma ação adaptativa, que altera a topologia, de forma que apenas expressões compatíveis sejam aceitas.

Ao ser executada a função adaptativa $\mathcal{F}$, com parâmetro $v.int$ ou $v.real$, insere-se no autômato uma transição que consome identificadores, assim classificados pelo analisador léxico, de forma que as expressões sejam compostas por elementos do mesmo tipo da variável destino.

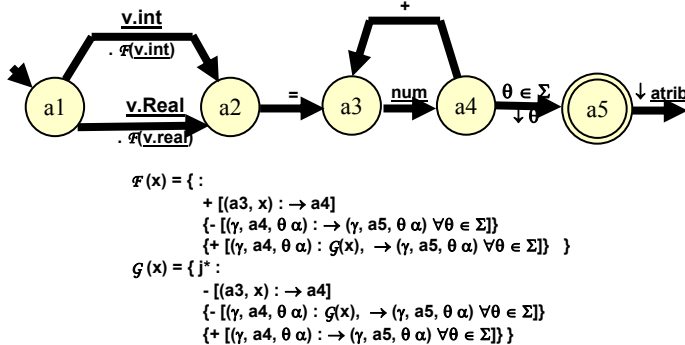


Figura 1: Autômato Adaptativo para tratamento dos tipos $v.int$ e $v.real$. Há duas funções adaptativas associadas ao modelo, descritas no exemplo [6].

Ao final do reconhecimento do comando de atribuição, essa transição acrescentada é removida por outra ação adaptativa (a ação $\mathcal{G}$), restaurando as condições iniciais. A Figura 1 ilustra o autômato adaptativo utilizado e representa as funções adaptativas necessárias para o tratamento dos tipos.

DE LÍNGUA NATURAL

Nesta seção, descrevem-se alguns dos trabalhos efetuados no Laboratório de Linguagens e Técnicas Adaptativas no que concerne ao Processamento de Língua Natural. Retomando a seção I, apresenta-se em seqüência: um etiquetador para Língua Portuguesa usando Tecnologia Adaptativa; a seguir, apresenta-se um estudo a respeito dos processos de análise sintática de línguas naturais; posteriormente, modelos para o tratamento de não-determinismo e ambigüidades na análise sintática de línguas naturais; e finaliza-se com a apresentação de um tradutor texto-voz baseado em Autômatos Adaptativos.

A. Análise Morfológica

Podem-se classificar os analisadores morfológicos nas seguintes categorias [4]:

- Baseados em Regras: as regras são construídas a *priori* por algum lingüista;
- Estatísticos:
 - Fortemente estatísticos: usam cálculos estatísticos mais complexos;
 - Fracamente estatísticos: baseados em contagem e em regras;
- Baseados em Exemplos Memorizados: os exemplos fornecem a base de aprendizagem a partir da qual o analisador evolui.

Os analisadores morfológicos apresentam idéias lingüísticas comuns, tais como: a) Lista de palavras associadas a categorias morfológicas, para fornecer informações sobre palavras conhecidas; b) Lista de sufixos (terminações de palavras) associados a categorias morfológicas, para fornecer informações sobre palavras desconhecidas; c) Contexto próximo do item lexical que se quer etiquetar (2 ou 3 etiquetas) para refinar a escolha de sua etiqueta.

O método empregado para a construção de um analisador morfológico baseado em Autômatos Adaptativos utiliza uma mescla do método Fracamente Estatístico com o método baseado em Exemplos Memorizados.

Assim, a sua arquitetura é similar à do método Fracamente Estatístico e armazenam-se exemplos, como o método baseado em Exemplos Memorizados, e sua realização física é feita com base nos Autômatos Adaptativos.

A arquitetura do etiquetador é uma arquitetura em três módulos, que realizam:

- Etiquetação inicial de palavras conhecidas;
- Etiquetação inicial de palavras desconhecidas;
- Refinamento contextual (uso de heurística).

Os pré-requisitos para que esse método funcione em uma determinada língua natural são: disponibilidade de um *corpus* com anotações morfológicas; as palavras dessa língua devem ter uma estrutura do tipo (Palavra = Prefixos + Radical + Sufixos); e ainda, a língua não deve ser aglutinante (com mais de um radical), como é o caso do alemão.

Exemplo: A palavra “A”

A maior dificuldade é a necessidade de um número bem

maior de etiquetas para a língua portuguesa, por exemplo, o *Corpus* Penn Treebank usa um conjunto de 36 etiquetas, a menos de pontuações, enquanto o *Corpus* Tycho Brahe usa 231 etiquetas.

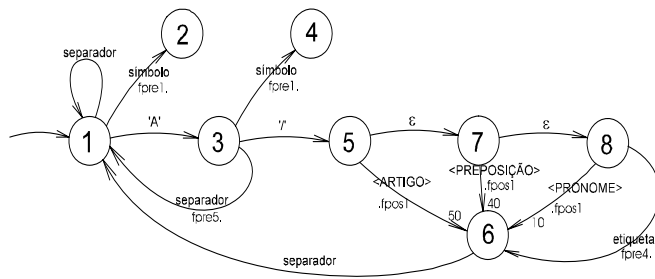


Figura 2: A palavra “A”, etiquetada como artigo, apareceu cinquenta vezes no *corpus* de treinamento, etiquetada como preposição, quarenta vezes, e, como pronome, dez vezes [4].

O primeiro experimento usou um *Corpus* de treinamento muito pequeno, para mostrar a viabilidade do projeto. Esse *Corpus* não faz mais parte do *corpus* Tycho Brahe e é composto de 1.812 itens lexicais e dividido em duas partes: Parte de treinamento: 1.684 itens lexicais; Parte de aplicação: 128 itens lexicais.

O tamanho do *Corpus* de treinamento usado foi praticamente insignificante: 1.684 itens lexicais anotados. Os resultados obtidos foram comparados aos de Alves e Villavicencio [apud [4]]:

- Taxa de acerto do experimento de 81,25%;
- Alves conseguiu a taxa de 78,28% com um *Corpus* de treinamento de 5.000 itens lexicais;
- Villavicencio alcançou 84,5% com um *Corpus* de treinamento de 14.000 itens lexicais.

B. Análise Sintática de Língua Natural

Historicamente, os formalismos associados às linguagens surgiram a partir da busca por métodos, técnicas e modelos formais a partir dos quais se pudesse gerar uma língua natural. Assim, cronologicamente, tem-se: Gramática de Estrutura de Frase, Gramática Livre de Contexto, Augmented Transition Network (redes ATN), Definite Clause Grammar, Formalismos baseados em Unificação e Formalismos baseados em Restrições (Lexical-functional grammar, Functional unification grammar, Generalized Phrase Structure Grammar, Head-Driven phrase-structure grammar, PATR-II), e Gramática de Dependência [15].

Como ilustração a respeito do trabalho desenvolvido, usa-se neste artigo a transposição efetuada em [15] de redes Augmented Transition Network (ATN) para Autômatos Adaptativos. Para isso, define-se a estrutura de uma rede ATN.

ATN - (R, Q, A, Q0, CS, RG)

R – Rótulo da rede

Q – Conjunto de estados

A – Conjunto de arcos (CAT, WRD, MEM, JUMP, PUSH, POP, VIR)

Q0 – Estado inicial da rede

CS – Conjunto de características sintáticas

RG – Conjunto de registradores

Essa estrutura permite descrever sintaticamente uma língua natural, embora não abranja adequadamente todas as regras gramaticais.

A partir de uma rede ATN qualquer, pode-se construir um Autômato Adaptativo que realiza as mesmas operações que a rede ATN, ou seja, se a rede realiza uma análise sintática correta, também o Autômato Adaptativo correspondente o fará. A Figura 3 mostra o mapeamento dos arcos ATN para as transições de um Autômato Adaptativo.

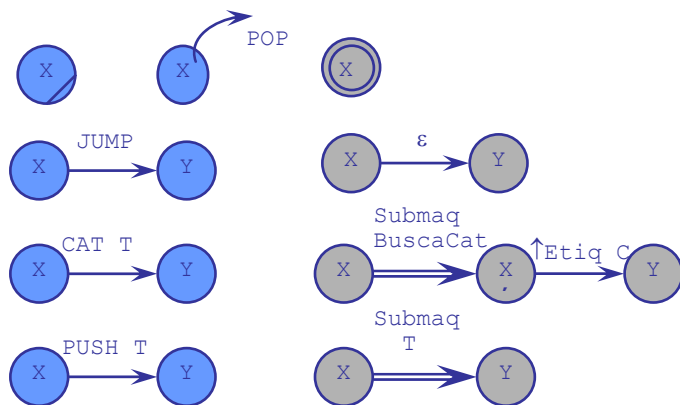


Figura 3: Mapeamento dos arcos ATN – Autômatos Adaptativos [15].

A partir do mapeamento de uma rede ATN, pode-se então construir um Autômato. Como exemplo, segue na Figura 4 a transformação da rede ATN em um Autômato Adaptativo [15].

Exemplo de Mapeamento ATN – AA:

```
(NP/
  (REGS DET NUMDET GENDET N NUM GEN PP)
  (NP/ (CAT DET T (SETR DET *) (SETR NUMDET (GETF
    NUMERO))
    (SETR GENDET (GETF GENERO)) (TO NP/DET))
    (CAT N T (SETR N *) (SETR NUM (GETF NUMERO))
    (SETR GEN (GETF GENERO)) (TO NP/N3)))
  ...
  (NP/PP2
    (PUSH PP/ T (SETR PP (APPEND PP *)) (TO NP/PP2))
    (POP T (BUILD (+ + + +) NUM GEN N PP))))
```

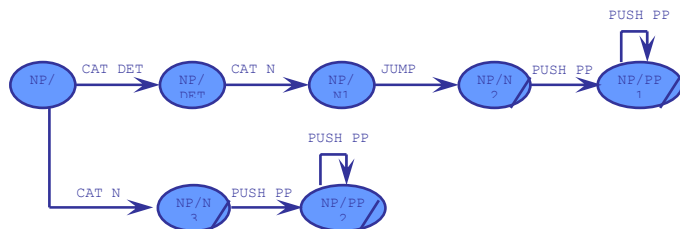


Figura 4: A rede ATN na parte superior da figura é mapeada para um Autômato Adaptativo usando-se a estrutura da Figura 3 [15].

Esse Mapeamento trouxe algumas contribuições, tais como:

- Constatação da viabilidade do uso de Autômatos Adaptativos na Representação de Línguas Naturais;
- Apresentação de utilização prática do Formalismo Adaptativo;
- Formas de mapeamento para Autômato Adaptativo e Gramática Adaptativa a partir de notação clássica para representação de língua natural;

- Implementação do mapeamento acima, com entrada compatível ao analisador e etiquetador morfológico disponível.

C. Tratamento Sintático de Língua Natural – Não-Determinismo e Ambigüidades

A resolução de não-determinismos e ambigüidades por meio de tratamento formal nem sempre é possível em linguagens livres de contexto. Em linguagens sensíveis ao contexto, essa situação torna-se ainda mais complexa.

O não-determinismo em uma língua natural ocorre quando duas ou mais construções sintáticas que possam ocorrer em um determinado ponto das sentenças da linguagem apresentam prefixo comum.

A ambigüidade é um fenômeno lingüístico no qual uma sentença pode ter duas ou mais interpretações válidas na mesma linguagem.

A proposta apresentada nesta seção é a de buscar o conjunto de todas as interpretações da sentença, tratando de maneira uniforme os não-determinismos e as ambigüidades, conforme discutido em [9,10].

1) Método para Tratamento de Não-Determinismo e Ambigüidade

Um reconhecedor adaptativo é elaborado de modo a permitir que as construções sem ambigüidade ou não-determinismo sejam aceitas da forma usual.

Ao se encontrar um não-determinismo ou ambigüidade, criam-se, para serem executadas em paralelo, cópias modificadas do autômato, cada qual capaz de reconhecer uma das possíveis construções sintáticas válidas.

O paralelismo é simulado através da execução alternada, passo a passo, assim, as diversas versões do autômato são executadas simultaneamente, sendo descartadas aquelas que não permitirem o prosseguimento do reconhecimento, e consideradas vitoriosas as que esgotarem a cadeia de entrada em situação final.

Havendo mais de uma máquina vitoriosa, há indicativo de ambigüidade na linguagem.

2) Construção do Reconhecedor

O reconhecedor adaptativo é construído a partir de uma gramática em notação de Wirth. O procedimento inicial usual em compiladores para construção de autômatos finitos ou de pilha estruturados segue [6].

Caso haja ambigüidades, caracterizadas por: $S \rightarrow S_1|S_2|...|S_n$, onde cada S_i é seqüência não obrigatória constituída minimamente de $\alpha_1\alpha_2...\alpha_m$, criam-se máquinas M_i , capazes de reconhecer as seqüências S_i ;

O Autômato Adaptativo apresenta uma submáquina principal que tem um único estado (origem e destino), consumindo cada um dos símbolos da cadeia de entrada. A essas transições são associadas as ações adaptativas $\mathcal{A}_1, ..., \mathcal{A}_m$.

Além dele, há o conjunto de submáquinas originais M_i , reconhecedoras das seqüências originais S_i , segundo a estrutura original de um processo de compilação [6].

Uma lista de apontadores completa a estrutura de reconhecimento e é usada para manter um conjunto de

transições em vazio. Aponta para o estado corrente de cada um dos autômatos de cada interpretação possível e possui uma ação adaptativa $\mathcal{B}$, que cria as cópias das submáquinas.

A lista de apontadores indica o estado corrente de cada interpretação e cada transição é simulada pela mudança dos apontadores (isso é efetuado pelas ações $\mathcal{A}_i$).

Se as transições simuladas alcançam um estado cujas transições possíveis são de chamada de submáquina, estas estão associadas a ações adaptativas de cópia ($\mathcal{C}_j$), que criam os ponteiros para cada cópia efetuada e os inserem na lista de ponteiros. Uma ação adaptativa $\mathcal{D}$ testa o estado corrente dos reconhecedores não descartados, eliminando aqueles que não lograrem sucesso no reconhecimento. Esse processo é baseado no conceito de cópia de modelos de autômato e execução simulada de modelos [9,10].

D. Uma Arquitetura Adaptativa para Processamento de Língua Natural

Uma das pesquisas desenvolvidas no Laboratório de Linguagens e Técnicas Adaptativas propôs uma arquitetura para o processamento de línguas naturais [9,10]. Essa arquitetura necessita de algumas definições para permitir a um lingüista o seu uso adequado em uma ferramenta [9,10].

Define-se e usa-se uma metalinguagem para descrição dos elementos lingüísticos de língua natural. A metalinguagem permite construir as estruturas lingüísticas da língua natural e seu posterior processamento computacional. Gera-se, a partir da descrição na metalinguagem, um transdutor baseado em Autômatos Adaptativos. O transdutor extrai as palavras do texto e as classifica segundo a categoria lexical. Posteriormente, realiza-se a análise sintática estática do texto e depois, a análise sintática completa do texto considerando dependência de contexto (concordância, regência, anáfora, pronomes, etc.). Finalmente, realiza-se a análise sintática restante considerando ambigüidade e não-determinismo (decorrentes de permutação de ordem dos componentes – geração de diversos caminhos válidos), usando-se as técnicas descritas na seção III.C [9,10].

E. Tradução Texto-Voz usando Autômatos Adaptativos

A pesquisa que resultou no tradutor texto-voz denominado *Voicer* foi conduzida inicialmente usando o AdapTools [12]. Posteriormente, a máquina virtual do dispositivo adaptativo (Autômato Adaptativo) foi refeita, tornada mais eficiente e diretamente voltada ao problema da vocalização [3]. O modelo de tradução para voz utilizado é o de vocalização de fonemas pré-gravados.

O tradutor texto-voz construído é composto por três blocos distintos: um analisador léxico, um analisador de sílabas e um analisador fonético.

1) Analisador Léxico

O analisador léxico segue um modelo padrão de compilação e não contém ações adaptativas. Entretanto, esse modelo de analisador separa as sílabas (fonemas).

2) Analisador das Sílabas

O analisador das sílabas prepara todas as alternativas

fonéticas, baseado nas sílabas recebidas do analisador léxico.

3) Analisador Fonético

O analisador fonético escolhe a alternativa mais correta entre as possibilidades levantadas pelo analisador das sílabas.

4) Configuração do Voicer

A Figura 5 ilustra a configuração inicial do autômato que realiza o Voicer. Os três analisadores estão mostrados na figura.

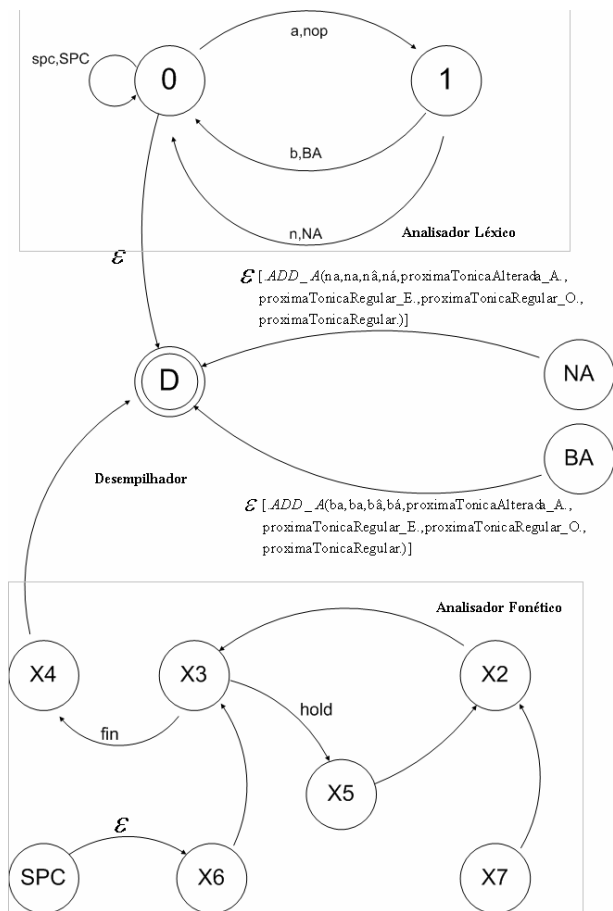


Figura 5: Configuração inicial do autômato que implementa o Voicer. Os três analisadores estão representados.

O tratamento sintático do contexto via adaptatividade ainda está sendo implementado, entretanto, o vocalizador já funciona adequadamente, podendo ser ainda melhorado.

Exemplo: Vocalização de “BANANA”

A vocalização da palavra “BANANA” provoca alterações no autômato que podem ser acompanhadas pela sequência de figuras abaixo.

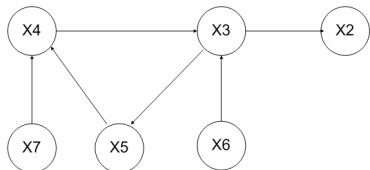


Figura 6: Configuração inicial do analisador fonético [3].

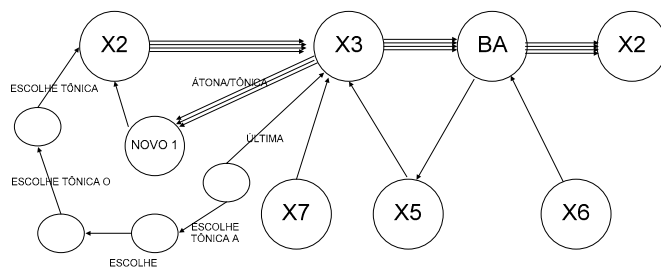


Figura 7: Configuração do analisador fonético após a sílaba ‘ba’ [3].

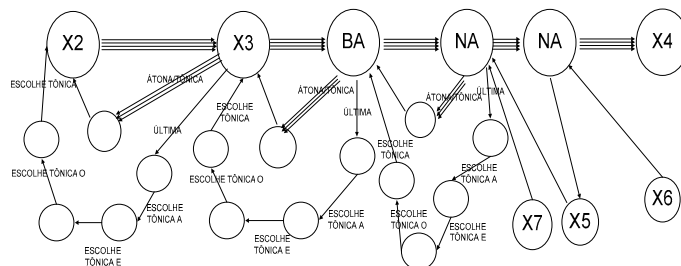


Figura 8: Configuração do analisador fonético após ‘banana’ [3].

Esse exemplo ilustrou a sequência de modificações estruturais enfrentadas pelo Autômato correspondente ao analisador fonético, de modo a vocalizar a sílaba tônica correta.

IV. CONCLUSÃO

Este artigo mostrou algumas das possibilidades em uso e em pesquisa no Laboratório de Linguagens e Técnicas Adaptativas no que tange ao processamento de língua natural [8]. Ilustrou e mostrou claramente a motivação para o uso de tecnologia adaptativa em situações nas quais há dependência de contexto.

Foram ainda ilustradas questões relativas ao tratamento de língua natural por meio computacional, mostrando modelos adaptativos para o seu processamento adequado. Embora os exemplos aqui apresentados tenham sido baseados em Autômatos Adaptativos, cumpre ressaltar que a construção de uma gramática para uma língua natural é também tema de pesquisa que vem sendo efetuada no Laboratório [2,5,8].

REFERÊNCIAS

- [1] A. Aho, R. Sehti e J. Ullman. *Compilers: Principles, techniques and tools*. Addison-Wesley, 2006.
- [2] C. A. Bravo Pariente. Gramáticas Livres de Contexto Adaptativas com Verificação de Aparência. *Tese de Doutorado*, USP, São Paulo, 2004.
- [3] A. J. B. Castro, D. A. Alfenas, D. P. Shibata, H. T. Soejima. Sintetizador Texto-Voz com Autômatos Adaptativos. Trabalho de Conclusão de Curso, USP, São Paulo, 2004.
- [4] C. E. D. Menezes e J. J. Neto. Um método para a construção de analisadores morfológicos, aplicado à língua portuguesa, baseado em autômatos adaptativos. *V PROPOR, Encontro para o Processamento Computacional de Português Escrito e Falado*, 19 a 22 de novembro de 2000, Atibaia, Brasil.
- [5] Iwai, M. K. Um formalismo gramatical adaptativo para linguagens dependentes de contexto. *Tese de Doutorado*, USP, São Paulo, 2000.
- [6] J. J. Neto. Contribuições à metodologia de construção de compiladores. *Tese de Livre Docência*, USP, São Paulo, 1993.

- [7] J. J. Neto. Adaptive Automata for Context -Sensitive Languages. *ACM SIGPLAN NOTICES*, Vol. 29, n. 9, pp. 115-124, September, 1994.
- [8] J. J. Neto. Autômatos em Engenharia de Computação – uma Visão Unificada. *Primera Semana de Ciencia y Tecnología de la Sociedad Chotana de Ciencias y la Red Mundial de Científicos Peruanos*, Junio 22-27, Ciudad de Chota, Perú, 2003.
- [9] J. J. Neto, and M. Moraes. Using Adaptive Formalisms to Describe Context-Dependencies in Natural Language. *Lecture Notes in Artificial Intelligence*. N.J. Mamede, J. Baptista, I. Trancoso, M. das Graças, V. Nunes (Eds.): *Computational Processing of the Portuguese Language 6th International Workshop, PROPOR 2003*, Volume 2721, Faro, Portugal, June 26-27, Springer-Verlag, 2003, pp 94-97.
- [10] J. J. Neto, e M. Moraes. Formalismo adaptativo aplicado ao reconhecimento de linguagem natural. *Anais da Conferencia Iberoamericana en Sistemas, Cibernética e Informática*, 19-21 de Julio, 2002, Orlando, Florida.
- [11] H. Lewis e C. Papadimitriou *Elements of the Theory of Computation*. Prentice-Hall, 1998.
- [12] H. Pistori e J. J. Neto. AdapTools: Aspectos de Implementação e Utilização *Boletim Técnico PCS*, Escola Politécnica, São Paulo, 2003.
- [13] R. L. A. Rocha e J. J. Neto. Autômato adaptativo, limites e complexidade em comparação com máquina de Turing. In: *Proceedings of the second Congress of Logic Applied to Technology – LAPTEC'2000*. São Paulo: Faculdade SENAC de Ciências Exatas e Tecnologia, p. 33-48, 2001.
- [14] R. W. Sebesta. *Programming language concepts*, Addison-Wesley, 2003.
- [15] C. Y. O. Taniwaki. Formalismos adaptativos na análise sintática de linguagem natural. *Dissertação de Mestrado*, USP, São Paulo, 2001.
- [16] C. D. Yang. Universal Grammar, statistics or both? *Trends in Cognitive Sciences*. Vol 8, n. 10, pp. 451-456, 2004.



Ricardo Luis A. Rocha é natural do Rio de Janeiro-RJ e nasceu em 29/05/1960. Graduou-se em Engenharia Elétrica modalidade Eletrônica na PUC-RJ, em 1982. É Mestre e Doutor em Engenharia de Computação pela EPUSP (1995 e 2000, respectivamente). Suas áreas de atuação incluem Tecnologias Adaptativas, Fundamentos de Computação e Modelos Computacionais. Dr. Rocha é membro da ACM (Association for Computing Machinery) e da SBC (Sociedade Brasileira de Computação).