Um método para a construção de etiquetadores morfológicos, aplicado à língua portuguesa, baseado em autômatos adaptativos

Carlos Eduardo Dantas de MENEZES
Dep. de Computação e Sistemas Digitais,
PCS - EPUSP, Universidade de São Paulo
Av. Prof. Luciano Gualberto, trav. 3, n. 158
CEP: 05508-900, São Paulo, Brasil
menezes@lsi.usp.br

João JOSÉ NETO
Dep. de Computação e Sistemas Digitais,
PCS - EPUSP, Universidade de São Paulo
Av. Prof. Luciano Gualberto, trav. 3, n. 158
CEP: 05508-900, São Paulo, Brasil
jjneto@pcs.usp.br

Abstract

This work intends to suggest a new method for the constructing of a morphological tagger designed to natural languages. In a large extent, this tagger is language-independent and may be trained by making use of a tagged corpus, which is now particularly applied to Portuguese texts.

Based on automatic learning principles, it was developed to acquire and infer, automaticaly, linguistic knowledge concerning the lexical and contextual aspects of a training corpus.

After being collected and inferred, the necessary information is changed into a coded structure, which was based on adaptive automata. This device is later used as a basis for the tagging of other texts.

In this project, adaptive automata has proven their adequacy for both the representation and the acquisition of knowledge on the mentioned aspects of the natural language and the logic of the heuristics employed to collect that required information.

Resumo

Este trabalho tem por objetivo propor um método de construção de um etiquetador morfológico, treinável com o uso de corpus anotado, que é independente de língua, mas que foi aqui testado com textos da língua portuguesa.

Trata-se de um sistema de aprendizado automático, que infere informações lingüísticas, relativas a aspectos lexicais e contextuais de todo um corpus de treinamento. Estas informações são armazenadas, codificadas com base em autômatos adaptativos, e posteriormente utilizadas para a tarefa de classificação ou etiquetação morfológica.

Os autômatos adaptativos mostraram-se adequados tanto para o fluxo de controle da heurística de aprendizado, como também para nele codificar todos os dados necessários.

1. Introdução

Um etiquetador morfológico tem por função associar a cada palavra uma etiqueta, que corresponda a sua categoria morfológica. Sua aplicação encontra-se em sistemas de tradução automática, em sistemas de auxílio à criação de corpora lingüísticos anotados, entre inúmeras outras tarefas do processamento de linguagens naturais (Isabelle e Bourbeau (1985), Marcus e outros (1993), TBCHP (1998)).

A principal dificuldade existente na tarefa da etiquetagem morfológica encontra-se em sua susceptibilidade à ambigüidade. Um etiquetador morfológico robusto deve levar em conta não apenas as informações lexicais da palavra a ser anotada, mas também informações a respeito do contexto em que esta palavra se encontra.

1.1 O estado da arte em etiquetadores morfológicos

Basicamente, pode-se dizer que quatro paradigmas ou métodos constituem o estado da arte na etiquetagem morfológica de textos em linguagem natural: o estatístico (Charniak (1993)), o que se utiliza de regras escritas manualmente (Koskenniemi (1997)), o baseado em regras inferidas automaticamente (Brill (1993)) e o com base em exemplos memorizados (Daelemans e outros (1996)). Todos eles conseguem uma taxa de acerto em torno dos 96% para textos na língua inglesa.

É possível observar idéias lingüísticas semelhantes em todos os paradigmas de etiquetadores morfológicos treináveis citados. Todos utilizam-se de três fontes de informação lingüística, extraídas de um corpus de treinamento:

- os sufixos de palavras, como parte do processo de inferência da etiqueta morfológica de palavras desconhecidas;
- uma lista de palavras associadas à categorias morfológicas (léxico), para fornecer informações sobre palavras conhecidas;
- contexto próximo do item lexical que se quer etiquetar (2 ou 3 etiquetas ao redor), para refinar a escolha de sua etiqueta.

2. Autômatos Adaptativos

Os autômatos adaptativos (AA) constituem um formalismo para a representação de linguagens dependentes de contexto (José Neto (1994)). A base estrutural de um AA é um autômato de pilha; o que os diferencia é que um AA pode ter, associado a cada uma de suas transições, funções adaptativas, anteriores e posteriores, conforme explanado adiante (figura 1).

As funções adaptativas são constituídas de um conjunto de **ações adaptativas elementares** que possibilitam modificar o autômato como decorrência da execução de uma transição, através do acréscimo e retirada de estados e transições. As **ações adaptativas elementares** podem ser de três tipos: Inspeção, Eliminação e Inserção.

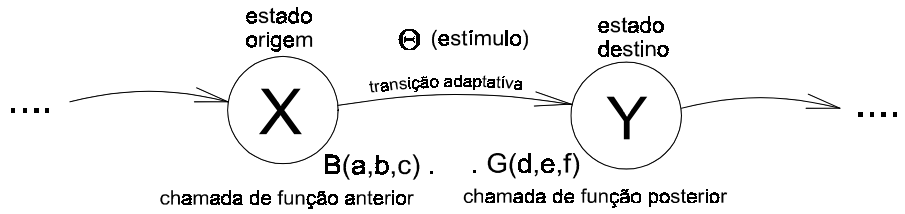

Figura 1 – Uma transição de Autômato Adaptativo (notação gráfica)

São estes dois últimos tipos de ações adaptativas elementares que dão aos AA o poder computacional para manipular linguagens dependentes de contexto (José Neto (1994)).

As **chamadas de funções adaptativas** podem ser de dois tipos: anterior (efetuada sempre antes de uma transição ocorrer) e posterior (efetuada sempre depois que a mudança de estado é realizada).

A característica de poder alterar sua própria topologia, peculiar aos autômatos adaptativos, faz com que eles sejam bastante adequados à modelagem de sistemas de aprendizado automático: um conjunto de exemplos poderia ser inserido em um AA (treinamento) na forma de novas transições; deste modo um AA pode incorporar conhecimento.

3. Proposta

3.1 Interpretador de Autômatos Adaptativos

Construiu-se um interpretador para um formalismo variante dos AAs, que será batizado de Autômatos Adaptativos E (“E” vem de Estendido), que consiste de um grande subconjunto dos autômatos adaptativos originais, com algumas extensões:

•É um AA que não faz uso de pilha.
•É estritamente determinístico (o autômato inicial deve ser determinístico e todas as alterações adaptativas devem mantê-lo determinístico).
•Possue uma extensão que fornece subsídios ao levantamento de dados estatísticos muito simples (apenas contagens), que são imprescindíveis para a implementação de algoritmos de aprendizado automático. Isto consiste em associar a cada transição uma variável inteira, automaticamente iniciada com o valor zero quando a transição é inserida, e criar mais dois tipos de **ações elementares**: **Soma um** (incrementa de um o valor da variável associada a uma transição) e **Comparação** (compara o valor da variável associada a uma transição com o valor de outra variável, associada a outra transição, ou então com um valor fixo).
•Foram definidos diversos estímulos, que serão usados pelos autômatos adaptativos E: **estímulo simples** (um caractere, uma cadeia de caracteres ou uma lista de cadeias de caracteres explícitos; exemplos: "A", "CONJ", "ADV/N/PREP"), **barra** (o caractere '/'), **separador** (o caractere espaço em branco ou um caractere que represente a mudança de linha), **símbolo** (qualquer caractere que não seja nem uma barra, nem um separador), **etiqueta** (qualquer cadeia de caracteres que não inclua separadores ou barras), **conj_tag** (uma lista qualquer de pelo menos duas **etiquetas**, separadas por uma barra). Cada um destes estímulos será considerado bem-sucedido quando houver na primeira posição da cadeia de entrada o referido caractere ou caracteres, sendo que se a transição à qual este estímulo está associado ocorrer, este mesmo estímulo será consumido da cadeia de entrada. Além destes ainda foram definidos: **lap** (que abrevia *look-ahead* de pertinência, é sempre acompanhado de um parâmetro que constitui um **estímulo simples**. Será bem-sucedido em duas situações: (1) quando houver na primeira posição da cadeia de entrada o **estímulo simples** que é o parâmetro do **lap** ou (2) quando houver na primeira posição da cadeia de entrada uma lista de etiquetas, e o parâmetro do **lap** for uma das etiquetas desta lista. De qualquer modo, nada será consumido da cadeia de entrada) e **vazio** (será bem-sucedido sempre e nada será consumido da cadeia de entrada).
•A ordem na qual os diferentes tipos de estímulos são examinados pelo interpretador é a seguinte: (1) estímulo simples, (2) lap, (3) símbolo, barra, etiqueta, conj_tag (todos com igual prioridade) e (4) vazio.

3.2 Especificação do etiquetador morfológico

Este trabalho propõe, entre outras coisas, um método para a construção de um etiquetador morfológico, que possa ser usado para um número muito grande de línguas (apesar que se propõe testá-lo apenas para a língua portuguesa), que seja treinável com o uso de corpus e que possibilite uma boa precisão na anotação.
Os pré-requisitos para que este etiquetador possa ser treinado numa determinada língua são os seguintes:
•disponibilidade de um corpus com anotações morfológicas;
•as palavras desta língua devem ter uma estrutura do tipo:

Palavra = Prefixos + Radical + Sufixos

•a língua não deve ser aglutinante (como é o caso do alemão, por exemplo), visto que este método ainda não leva em conta a existência de itens lexicais com mais de um radical.

A arquitetura básica do etiquetador morfológico treinável proposto neste trabalho segue o publicado por E. Brill, que divide-se em três módulos (Brill (1993)): o primeiro, que cuida da etiquetação inicial de palavras conhecidas, o segundo, que cuida da etiquetação inicial de palavras desconhecidas, e um terceiro e último, que promove um refinamento contextual (figura 2).

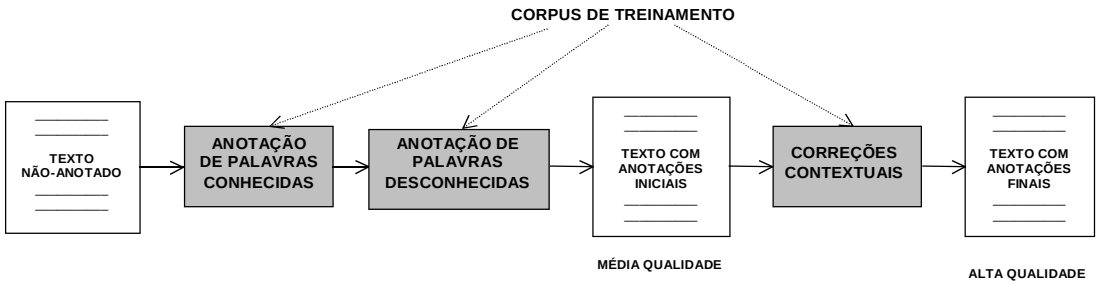

Figura 2 – Visão macroscópica do etiquetador morfológico proposto

Cada um dos módulos armazenará informações extraídas de um corpus de treinamento, e, a partir destas informações, procederá a etiquetação sem a inferência de regras explícitas, o que está mais próximo da proposta feita em Daelemans e outros (1996). Usar-se-ão autômatos adaptativos (AA) como base de implementação e como estrutura de dados para o armazenamento das informações necessárias a cada módulo (José Neto (1994)).

3.2.1 Primeiro módulo: obtenção da etiqueta mais provável para as palavras conhecidas

A estrutura de dados concebida como base para este módulo é a de uma árvore *n*-ária de letras, utilizada para armazenar o léxico, contendo uma lista ligada associada a cada uma de suas folhas (a qual representa o final de uma seqüência completa que compõe um item lexical). Esta lista é utilizada para armazenar as várias etiquetas morfológicas possíveis, em ordem decrescente de freqüência de aparecimento. Uma vantagem, inerente a esta estrutura em forma de árvore, é que ocorre naturalmente uma compressão do tamanho da base de dados pelo fato de todos os prefixos serem armazenados apenas uma vez na estrutura. A rigor, considerando-se todas as transições deste autômato, inclusive aquelas que processam os separadores do texto de entrada, ele constitue um grafo orientado cíclico.

Após a fase de treinamento, um texto sem anotações pode ser fornecido ao autômato e este determinará as etiquetas mais prováveis, em ordem decrescente de freqüência, para cada palavra que tenha aparecido no corpus de treinamento (ou seja, uma palavra conhecida); caso uma palavra deste texto não tenha aparecido neste corpus (palavra desconhecida), ela não receberá etiqueta alguma.

A figura 3 mostra como ficaria este autômato após ser alimentado por um corpus de treinamento hipotético que só apresenta a palavra “A”, algumas vezes etiquetada como ARTIGO, outras como PREPOSIÇÃO e ainda outras vezes como PRONOME.

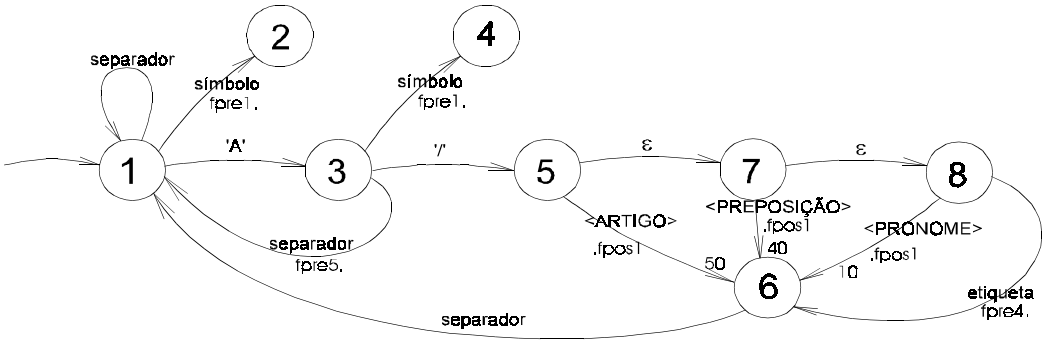

Figura 3 – Autômato adaptativo usado na obtenção da etiqueta mais provável para palavras conhecidas

3.2.2 Segundo módulo: etiqueta para palavras desconhecidas, com base em sufixos

Com base nas últimas letras dos itens lexicais encontrados no corpus de treinamento e nas etiquetas morfológicas associadas a cada um deles, este módulo infere um mapeamento que é usado na etiquetação de itens lexicais que nunca apareceram no corpus de treinamento (palavras desconhecidas).

A heurística por trás deste módulo tem um embasamento lingüístico: é sabido que, nas línguas cujas palavras apresentam a estrutura PREFIXO + RADICAL + SUFIXO, o sufixo de uma palavra tem uma forte correlação com a sua categoria morfológica.

Pode-se ter uma idéia geral do funcionamento deste módulo através da sua arquitetura, mostrada na figura 4.

Em princípio, deve-se fazer um pré-processamento no corpus de treinamento (elementos INVERSÃO, PODA e ALTERNÂNCIA na figura 4), para que o mesmo seja reduzido a apenas terminações de palavras com as respectivas etiquetas. Optou-se por reduzir cada item lexical a apenas suas três últimas letras (elemento PODA); é verdade que na língua Portuguesa há sufixos menores e maiores que 3 letras, contudo, a escolha deste número é arbitrária e este valor pode ser facilmente alterado, pois é um parâmetro do pré-processamento.

Deve-se também levar em conta que este módulo propricia uma forma de **extrapolação** quando faz uma comparação de sufixos, ou seja, qualquer comparação a ser feita não necessita ser exata, podendo ser parcial. Assim, uma palavra que tenha o sufixo “mente” receberia a etiqueta ADV (advérbio), mesmo que o módulo de palavras desconhecidas tenha associado apenas as últimas três letras “nte” a esta etiqueta.

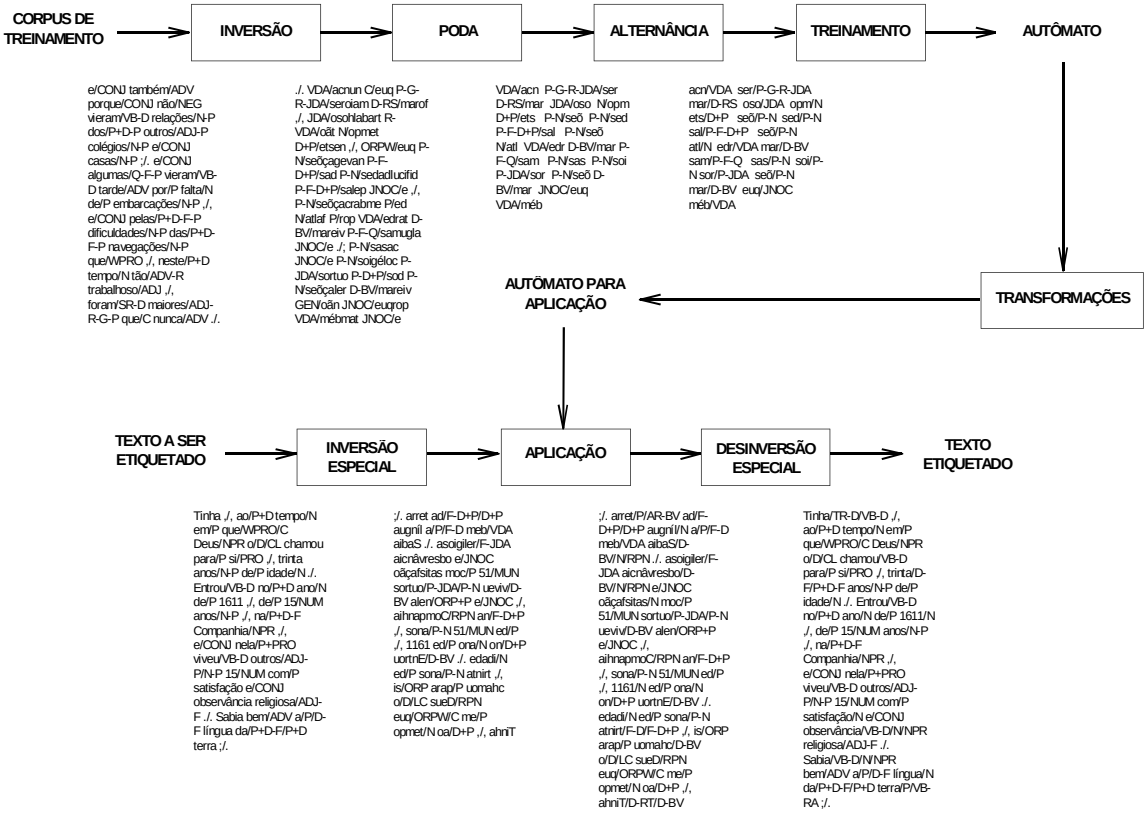

Figura 4 – Arquitetura do segundo módulo do etiquetador

Para ilustrar o processo de treinamento, usar-se-á um pequeno corpus hipotético, mostrado abaixo, que é, de fato, um subconjunto do corpus real. Nota-se que este já foi pré-processado.

```
ava/D-BV    ava/D-TE    aso/F-JDA
```

Este corpus traduz os seguintes fatos: que o sufixo “ava” (talvez advindo de palavras como “trabalhava”, “estava”, por exemplo) pode estar associado às etiquetas “VB-D” e “ET-D”, e que o sufixo “osa” (por exemplo, das palavras “carinhosa” e “custosa”) pode estar associado à etiqueta “ADJ-F”.

A figura 5 descreve o autômato inicial deste segundo módulo, o qual tem apenas dois estados e duas transições.

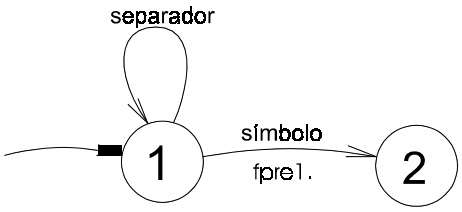

Figura 5 – Autômato inicial usado no segundo módulo do etiquetador

As etiquetas “VB-D” e “ET-D”, que são ambas associadas ao sufixo “ava”, são incorporadas ao autômato, conforme observa-se na figura 6. O crescimento do autômato, neste ponto, poderia acontecer através de uma dentre três funções adaptativas diferentes (fpre1, fpre4 ou fpre2), dependendo se o próximo sufixo não compartilha qualquer letra com os sufixos já armazenados (é o caso de fpre1), se este compartilha alguma letra, mas não todas, com um dos sufixos armazenados (fpre4) ou se compartilha todas as letras de um sufixo (é o caso de fpre2, que só faria sentido se houvesse algum sufixo com mais de três letras).

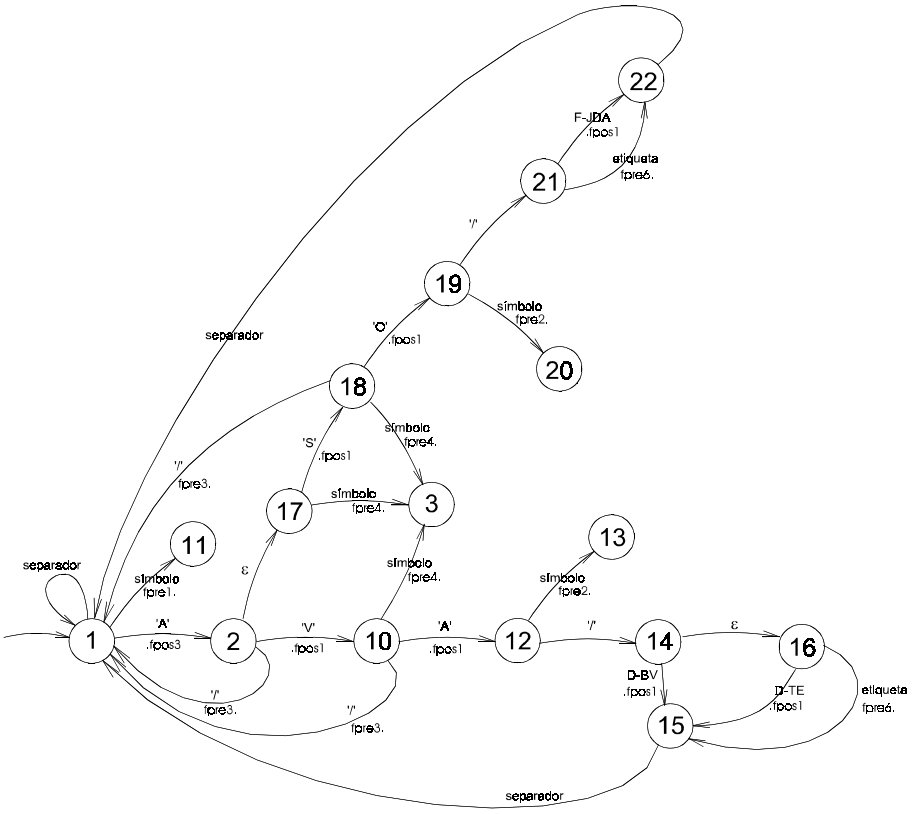

Figura 6 – Treinamento do segundo módulo do etiquetador: duas etiquetas são associadas ao sufixo “AVA” e uma ao sufixo “OSA"

Nesta mesma figura, vê-se também o modo pelo qual o autômato aprende o terceiro exemplo. Pode-se observar que todos os possíveis sufixos e seus relacionamentos com as correspondentes categorias morfológicas são automaticamente inferidos, gerando-se assim mapeamentos entre os sufixos e as etiquetas morfológicas mais prováveis das palavras correspondentes.

Para que o autômato do segundo módulo possa ser usado no processo de etiquetação, é proposto um conjunto de transformações neste que podarão as transições ligadas ao seu crescimento (processo de aprendizagem) e acrescentarão novas. Estas novas transições encarregar-se-ão de encontrar a etiqueta mais provável para o respectivo sufixo.

Assim que essas transformações forem efetuadas, o autômato já poderá ser usado para o processo de etiquetação. Tornou-se necessário um pós-processamento para a saída gerada por este autômato, com o objetivo de desfazer as inversões realizadas anteriormente. Isto pode ser observado, macroscopicamente, no elemento DESINVERSÃO ESPECIAL da figura 4.

3.2.3 Terceiro módulo: refinador contextual

Os dois módulos anteriores cuidam apenas de informações meramente lexicais extraídas de um corpus. Já o terceiro módulo serve como um refinador do serviço que os dois primeiros prestam. Ele é responsável por escolher, dentre as várias etiquetas possíveis para uma dada palavra, aquela que mais se adapte ao contexto em que esta palavra se encontra.

Este módulo é treinado com base em informações referentes à seqüência relativa em que as anotações morfológicas se acham no corpus de treinamento. Um pré-processamento é realizado no corpus de treinamento de modo a restar apenas as etiquetas (figura 7).

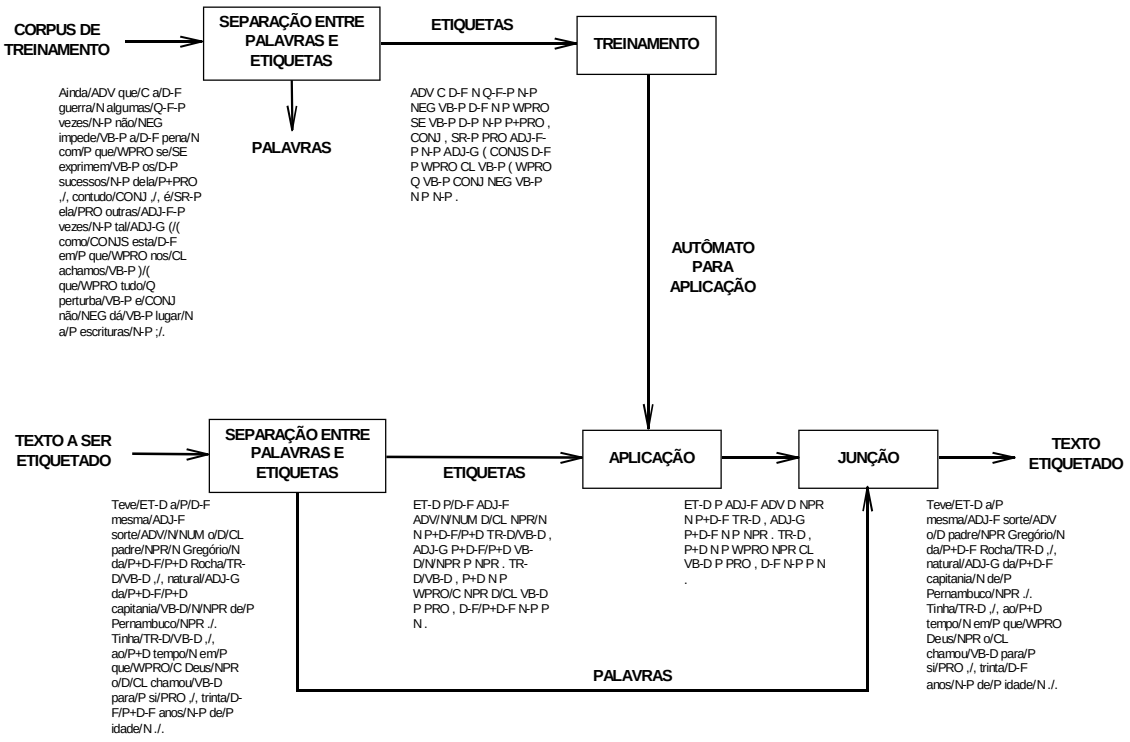

Figura 7 – Arquitetura do terceiro módulo do etiquetador

A figura 8 mostra o autômato inicial usado neste terceiro módulo do etiquetador. Este autômato consiste de apenas duas transições e três estados.

Supor-se-á o corpus hipotético abaixo para ilustrar a heurística de treinamento adotada.

P SR ADV-R CONJ ADV-R .... P N VB-D .... P N SR

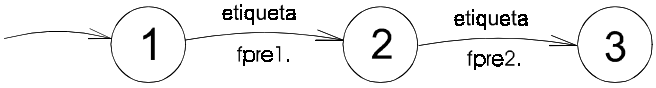

Figura 8 – Autômato inicial usado no terceiro módulo (refinador contextual) do etiquetador

A idéia central do método baseia-se na utilização de uma janela de três posições. Percorre-se com ela a seqüência de etiquetas previamente extraída do corpus de treinamento; a primeira posição da janela refere-se a uma etiqueta já consumida anteriormente, mas que é memorizada; a segunda, refere-se à etiqueta que está sendo consumida na ocasião, e a terceira, à etiqueta seguinte, que é apenas consultada, sem ser consumida (um *look-ahead*), conforme ilustrado na figura 9.

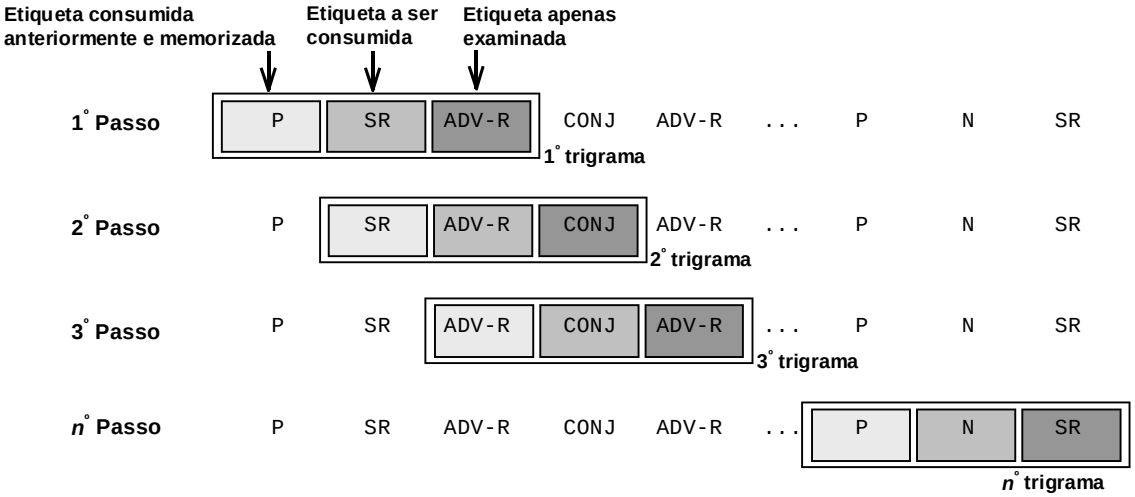

Figura 9 – Trigramas sendo armazenados durante treinamento do terceiro módulo do etiquetador

Esta janela desloca-se um passo por vez, sendo que a cada passo da janela, o correspondente trigrama é considerado.

Durante o primeiro passo, são encontradas as etiquetas P SR ADV-R. As duas primeiras etiquetas foram consumidas, enquanto que a terceira foi apenas examinada. Após os dois últimos trigramas do exemplo (P N VB-D e P N SR) serem examinados (figura 10), o autômato mantem uma estrutura de dados similar a uma árvore.

A heurística de aplicação do conhecimento contextual adquirido é resumido na figura 11. Pressupõe-se que o processo se inicie a partir de uma etiqueta não ambígua; a etiqueta seguinte, a qual será chamada de Foco, é a que será refinada, tendo em vista as etiquetas anterior e posterior (esta última pode ser ambígua ou não). Portanto, será escolhida uma dentre as várias etiquetas possíveis, de acordo com o contexto, para substituir o Foco.

Olhando-se para a primeira janela da figura 11 (1° Passo), percebe-se que existem 6 ($1 \times 3 \times 2$) possibilidades de trigramas sem ambigüidades, conforme listado a seguir:

P	ADJ	VB-D
P	ADJ	ADV
P	**N**	**VB-D**
P	N	ADV
P	SR	VB-D
P	SR	ADV

Suponha-se que apenas o trigrama ressaltado acima (P N VB-D) apareça no corpus de treinamento. Seria, então, natural esperar que o módulo de refinação contextual optasse pela etiqueta N para substituir a etiqueta ambígua ADJ/N/SR.

Estas decisões são tomadas dinamicamente pelo autômato através da montagem de novas transições: uma transição cujo estímulo é a própria etiqueta ambígua, e, em sua seqüência, um conjunto de transições com estímulo **lap**; estes estímulos testarão a etiqueta seguinte, que possivelmente é ambígua, sem entretanto consumi-la, transitando pela primeira que for bem-sucedida. Esta transição propiciará a escrita, na cadeia de entrada, da etiqueta refinada contextualmente, que no caso desta simulação seria a etiqueta N.

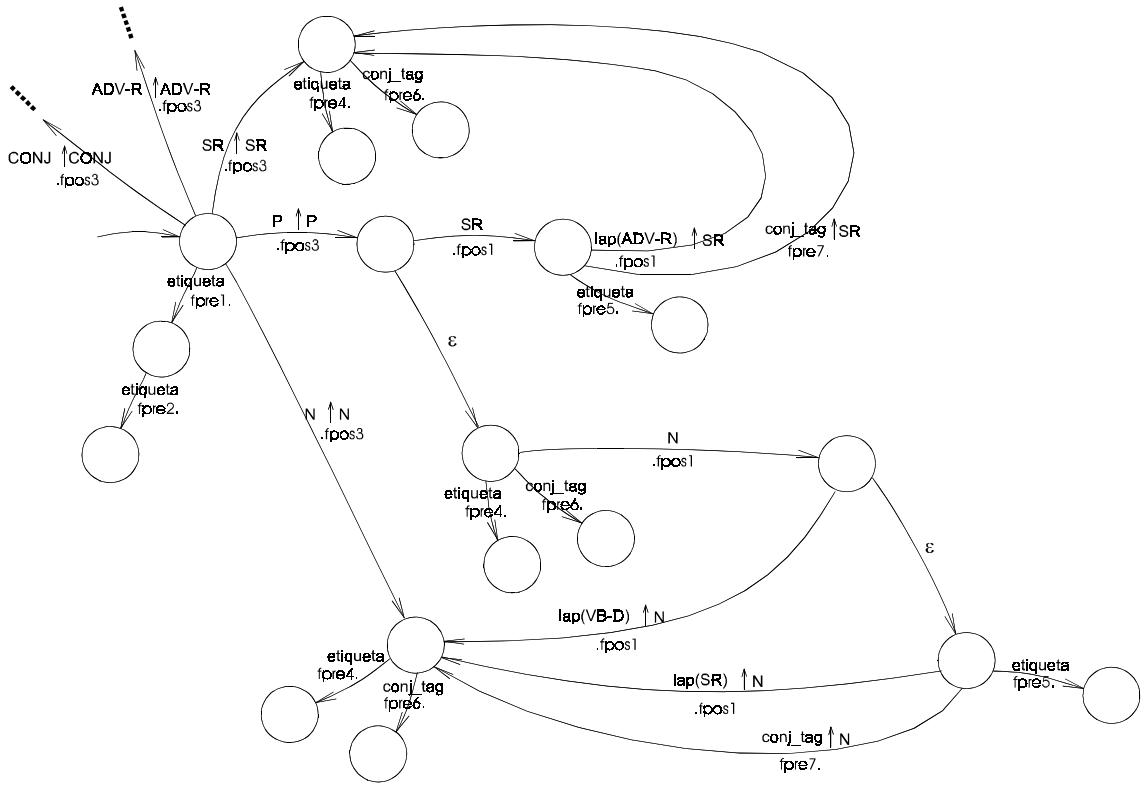

Figura 10 – Autômato durante a fase de treinamento do refinador contextual

A última das transições deste conjunto tem como estímulo a chamada Opção *Default*, a qual é apenas uma transição em vazio que envia para a cadeia de entrada a etiqueta mais provável dentre as componentes do Foco (etiqueta ambígua a ser corrigida), sem levar em consideração o contexto. No referido exemplo, esta é a etiqueta ADJ. Ou seja, se o contexto não oferecer informações suficientes (talvez por causa do treinamento com um corpus de tamanho pequeno, pouco significativo), opta-se por utilizar as informações lexicais vindas dos módulos anteriores (o de palavras conhecidas e o de palavras desconhecidas), considerando-se apenas a etiqueta mais provável.

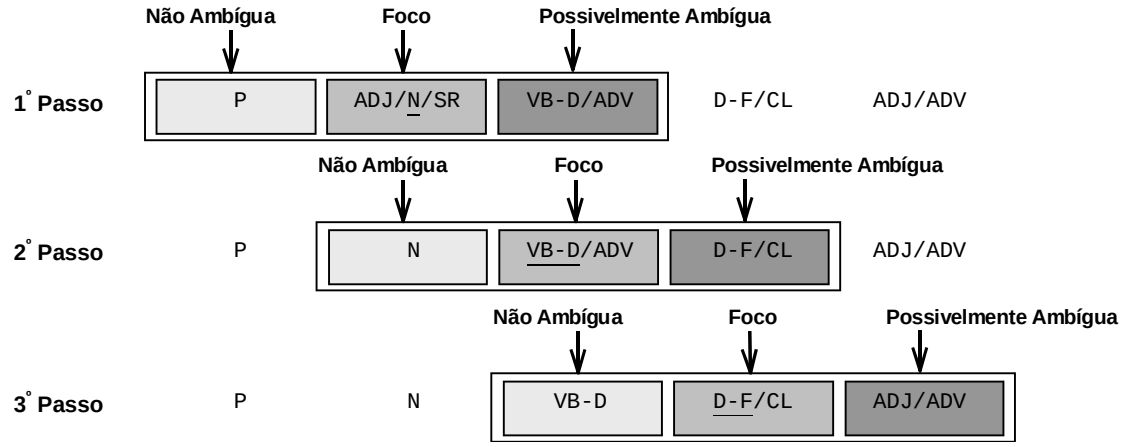

Figura 11 – Janela de três posições para resolver as ambigüidades pelo contexto

4. Experimentos Realizados

A dificuldade central desta tarefa, em comparação com a anotação morfológica em línguas como o inglês, reside no fato de que há a necessidade de um número bem maior de etiquetas para representar a maior riqueza morfológica da língua portuguesa (o corpus *Penn Treebank*, com textos em inglês, usa um conjunto de 36 etiquetas morfológicas, a menos de pontuações, enquanto que o corpus Tycho Brahe, com textos em português, usa 231 etiquetas. O método para a etiquetação morfológica proposto neste trabalho não é afetado por esta dificuldade.

O primeiro experimento realizado é um tanto quanto limitado do ponto de vista de validade prática, no entanto, tem uma função conceitual importante. Os diversos módulos foram treinados com o uso de um trecho que não faz mais parte do corpus Tycho Brahe (foi usado o que era disponível na época da realização do experimento), composto de 1.812 palavras e dividido em duas partes: corpus de treinamento, contendo 1.684 itens lexicais (palavras e pontuações) e corpus de aplicação, com 128 itens lexicais (TBCHP (1998)).

A tabela 1 resume o desempenho do etiquetador morfológico, nos seus diversos módulos.

Taxa de acerto		
1º módulo	**2º módulo**	**3º módulo (final)**
70,31%	81,25%	82,81%

Tabela 1 – Resumo do desempenho do etiquetador morfológico no 1º experimento

A taxa de acerto obtida (82,81%) é comparável ao relatado em Finger e Alves (1999), que conseguiu uma taxa de 78,28%, com um corpus de treinamento de 5.000 palavras, e ao relatado em Villavicencio e outros (1995), que alcançou 84,5%, com um corpus de treinamento de 14.000 palavras.

Como já era esperado, o aumento do corpus de treinamento propiciou um considerável aumento nesta taxa de acerto. O segundo experimento realizado já é mais abrangente e confiável, em aspecto prático. Os três módulos foram treinados com o uso de um texto de António das Chagas (1631-1682), que faz parte do corpus Tycho Brahe, e que é composto de 57.425 palavras, divididas da seguinte forma: corpus de treinamento contendo 51.017 itens lexicais e corpus de aplicação com 6.408 itens lexicais.

Conseguiu-se um bom desempenho final neste experimento, chegando à casa dos 90%, o que pode ser considerado bom, havendo, contudo, espaço para melhoras. Deve-se levar em conta que E. Brill começou a fazer experimentos que produziram resultados práticos com um corpus de

45.000 palavras (Brill (1993)); Daelemans e outros (1996) argumenta que o método baseado em exemplos memorizados começa a produzir resultados satisfatórios a partir de um corpus com 300.000 palavras.

5. Conclusão

Sem dúvida nenhuma, este trabalho constitue uma constatação significativa da adequação dos AA para a representação e manipulação de conhecimento da área de PLN (processamento de linguagens naturais). Mostrou sua viabilidade especialmente para a modelagem de algoritmos de aprendizado automático. Também deve ser ressaltado que o etiquetador para a língua portuguesa gerado é a primeira aplicação prática de larga escala baseada nos AA, mostrando que estes são dispositivos simples, elegantes, eficientes e treináveis.

Sob o olhar da lingüística computacional ou do PLN, foi construída uma ferramenta que propricia a etiquetagem morfológica de textos livres, com taxa de acerto comparável às dos paradigmas que representam o estado-da-arte na área, e com algumas vantagens, como:

- A complexidade computacional do treinamento e da aplicação dos três módulos que formam o etiquetador morfológico é independente do número de etiquetas e linear com respeito à cadeia de entrada. Isto é uma grande vantagem deste método proposto, em relação ao de E. Brill (cuja fase de treinamento tem dependência polinomial com relação à quantidade de etiquetas) e em relação ao de Finger e Alves (os quais adaptaram o etiquetador de Brill para a língua portuguesa) pois não necessita de um módulo adicional, com um conjunto de regras escritas manualmente, para dar ao etiquetador condições de manipular uma grande quantidade de etiquetas (Brill (1993), Finger e Alves (1999)).
- A possibilidade de explicar ou justificar uma decisão tomada, com base na maior proximidade de um determinado exemplo memorizado.
- Como o autômato adaptativo presente no etiquetador comporta-se de modo praticamente igual a um autômato finito, depois da fase de treinamento (já que as ações adaptativas nesta fase não são tão usadas), o desempenho da implementação pode estar muito perto do melhor possível, que seria o de um autômato finito (Roche e Schabes (1997)).

Referências

Brill, E. (1993) **A corpus-based approach to language learning**. Thesis (PhD) - Department of Computer and Information Science of the University of Pennsylvania, Philadelphia, 154 p.

Charniak, E. (1993) **Statistical language learning**. MIT Press.

Daelemans, W.; Zavrel, J.; Berck, P.; Gillis, S. (1996) MBT: A memory-based part of speech tagger-generator. In **Proceedings WVLC**, Copenhagen.

Finger, M.; Alves, C. (1999) Etiquetagem do Português Clássico Baseada em Córpora. In **Proceedings of IV Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR99).**, Évora, Portugal.

Isabelle, P.; Bourbeau, L. (1985) TAUM-AVIATION: its technical features and some experimental results. **Computational Linguistics**, v.11, n.1, p.18-27.

José Neto, J. (1994) Adaptative automata for context-dependent languages. **ACM SIGPLAN Notices**, v.29, n.9, p.115-24.

Koskenniemi, K. (1997) Representations and finite-state components in natural language, p.99-116. In ROCHE, E.; SCHABES, Y. (Eds.) – **Finite-state language processing**. MIT Press.

Marcus, M.; Santorini, B.; Marcinkiewicz, M. (1993) Building a large annotaded corpus of English: the Penn Treebank. **Computational Linguistics**, v. 19, n.2, p.313-30.

Roche, E.; Schabes, Y. (1997) Deterministic part-of-speech tagging with finite-state transducers, p.205-39. In ROCHE, E.; SCHABES, Y. (Eds.) – **Finite-state language processing**. MIT Press.

TBCHP (1998) Tycho Brahe Parsed Corpus of Historical Portuguese. **Instituto de Estudos da Linguagem**, UNICAMP, SP, http://www.ime.usp.br/~tycho/corpus.

Villavicencio, A.; Marques, N.; Lopes, G.; Villavicencio, F. (1995) Part-of-Speech Tagging for Portuguese

Texts Introduction, p.323-32. In WAINER, J.; CARVALHO, A. (Eds.) – Lecture Notes in Artificial Intelligence 991 – Advances in Artificial Intelligence – 12th Brazilian Symposium on Artificial Intelligence, SBIA'95, Campinas, Brazil, Springer-Verlag.