Etiquetamento Morfológico Usando Tecnologia Adaptativa

D. Padovani, A. T. Contier e J. José Neto

***Resumo*— Este trabalho apresenta uma revisão dos conceitos de Tecnologia Adaptativa e de Processamento da Linguagem Natural, e descreve a arquitetura do reconhecedor gramatical Linguístico, com ênfase no módulo Identificador Morfológico. Em seguida, são apresentados experimentos nos quais as submáquinas adaptativas de etiquetamento morfológico foram avaliadas, comparando seu desempenho ao do etiquetamento realizado sem o uso de tecnologia adaptativa. Por fim, são apresentadas as conclusões dos experimentos e as linhas de continuidade da pesquisa.**

***Palavras Chave* — Autômatos Adaptativos, Processamento de Linguagem Natural, Reconhecedores Gramaticais, Gramáticas Livres de Contexto, Gramáticas Adaptativas.**

I. Autômatos Adaptativos

O AUTÔMATO adaptativo é uma máquina de estados à qual são impostas sucessivas alterações resultantes da aplicação de ações adaptativas associadas às regras de transições executadas pelo autômato [1]. Dessa maneira, estados e transições podem ser eliminados ou incorporados ao autômato em decorrência de cada um dos passos executados durante a análise da entrada. De maneira geral, pode-se dizer que o autômato adaptativo é formado por um dispositivo convencional, não adaptativo, e um conjunto de mecanismos adaptativos responsáveis pela auto modificação do sistema.

O dispositivo convencional pode ser uma gramática, um autômato, ou qualquer outro dispositivo que respeite um conjunto finito de regras estáticas. Este dispositivo possui uma coleção de regras, usualmente na forma de cláusulas if-then, que testam a situação corrente em relação a uma configuração específica e levam o dispositivo à sua próxima situação. Se nenhuma regra é aplicável, uma condição de erro é reportada e a operação do dispositivo, descontinuada. Se houver uma única regra aplicável à situação corrente, a próxima situação do dispositivo é determinada pela regra em questão. Se houver mais de uma regra aderente à situação corrente do dispositivo, as diversas possíveis situações seguintes são tratadas em paralelo e o dispositivo exibirá uma operação não determinística. Os mecanismos adaptativos são formados por três tipos de ações adaptativas elementares: consulta (inspeção do conjunto de regras que define o dispositivo), exclusão (remoção de alguma regra) e inclusão (adição de uma nova regra).

Autômatos adaptativos apresentam forte potencial de aplicação ao processamento de linguagens naturais, devido à facilidade com que permitem representar fenômenos linguísticos complexos tais como dependências de contexto. Adicionalmente, podem ser implementados como um formalismo de reconhecimento, o que permite seu uso no pré-processamento de textos para diversos usos, tais como: análise sintática, verificação de sintaxe, processamento para traduções automáticas, interpretação de texto, corretores gramaticais e base para construção de sistemas de busca semântica e de aprendizado de línguas auxiliados por computador.

Diversos trabalhos confirmam a viabilidade prática da utilização de autômatos adaptativos para processamento da linguagem natural. É o caso, por exemplo, de [2], que mostra a utilização de autômatos adaptativos na fase de análise sintática; [3] que apresenta um método de construção de um analisador morfológico e [4], que apresenta uma proposta de autômato adaptativo para reconhecimento de anáforas pronominais segundo algoritmo de Mitkov.

II. Processamento da Linguagem Natural: Revisão da Literatura

O processamento da linguagem natural requer o desenvolvimento de programas que sejam capazes de determinar e interpretar a estrutura das sentenças em muitos níveis de detalhe. As linguagens naturais exibem um intricado comportamento estrutural visto que são profusos os casos particulares a serem considerados. Uma vez que as linguagens naturais nunca são formalmente projetadas, suas regras sintáticas não são simples nem tampouco óbvias e tornam, portanto, complexo o seu processamento computacional. Muitos métodos são empregados em sistemas de processamento de linguagem natural, adotando diferentes paradigmas, tais como métodos exatos, aproximados, pré-definidos ou interativos, inteligentes ou algorítmicos [5]. Independentemente do método utilizado, o processamento da linguagem natural envolve as operações de análise léxico-morfológica, análise sintática, análise semântica e análise pragmática [6]. A análise léxico-morfológica procura atribuir uma classificação morfológica a cada palavra da sentença, a partir das informações armazenadas no léxico [7]. O léxico ou dicionário é a estrutura de dados contendo os itens lexicais e as informações correspondentes a estes itens. Entre as informações associadas aos itens lexicais, encontram-se a categoria gramatical do item, tais como substantivo, verbo e adjetivo, e os valores morfossintático-semânticos, tais como gênero, número, grau, pessoa, tempo, modo, regência verbal ou nominal. Na etapa de análise sintática, o analisador verifica se uma sequência de palavras constitui uma frase válida da língua, reconhecendo-a ou não. O analisador sintático faz uso de um léxico e de uma gramática, que define as regras de combinação dos itens na formação das frases. Nos casos nos quais há a necessidade de interpretar o significado de um

texto, a análise léxico-morfológica e a análise sintática não são suficientes, sendo necessário realizar um novo tipo de operação, denominada análise semântica [7]. Na análise semântica procura-se mapear a estrutura sintática para o domínio da aplicação, fazendo com que a estrutura ganhe um significado. O mapeamento é feito identificando as propriedades semânticas do léxico e o relacionamento semântico entre os itens que o compõe [8]. Já a análise pragmática procura reinterpretar a estrutura que representa o que foi dito para determinar o que realmente se quis dizer. Inserem-se nessa categoria as relações anafóricas, correferências, determinações, focos ou temas, dêiticos e elipses [9]. Em [10] são apresentados trabalhos de pesquisas em processamento de linguagem natural para a Língua Portuguesa, tais como o desenvolvido pelo Núcleo Interinstitucional de Linguística Aplicada (NILC) no desenvolvimento de ferramentas para processamento de linguagem natural; o projeto VISL – Visual Interactive Syntax Learning, sediado na Universidade do Sul da Dinamarca, que engloba o desenvolvimento de analisadores morfossintáticos para diversas línguas, entre as quais o português; e o trabalho de resolução de anáforas desenvolvido pela Universidade de Santa Catarina. A tecnologia adaptativa também tem contribuído com trabalhos em processamento da linguagem natural. Em [11], são apresentadas algumas das pesquisas desenvolvidas pelo Laboratório de Linguagens e Tecnologia Adaptativa da Escola Politécnica da Universidade de São Paulo: um etiquetador morfológico, um estudo sobre processos de análise sintática, modelos para tratamento de não determinismos e ambiguidades, e um tradutor texto voz baseado em autômatos adaptativos.

I. O Reconhecedor Gramatical Linguístico

O Linguístico é uma proposta de reconhecedor gramatical composto de cinco módulos sequenciais que realizam cada qual um processamento especializado, enviando o resultado obtido para o módulo seguinte, tal como ocorre em uma linha de produção, até que o texto esteja completamente analisado [12]. A Fig. 1 apresenta a estrutura geral do Linguístico.

Este artigo procura focar o terceiro módulo, chamado Identificador Morfológico, detalhando sua arquitetura e apresentando um experimento no qual o módulo é testado com *tokens* da Língua Portuguesa, comparando os resultados com os obtidos sem o uso de tecnologia adaptativa. O Identificador Morfológico está diretamente relacionado a dois outros módulos do Linguístico, o Sentenciador e o Tokenizador. O Sentenciador é o módulo que recebe textos e os dividem em sentenças, usando, para isso, expressões regulares que aplicam regras de pontuação e de desambiguação de palavras abreviadas e de palavras compostas. O Tokenizador é o módulo que recebe as sentenças identificadas pelo Sentenciador e as divide em *tokens*, considerando, neste processo, abreviaturas, valores monetários, horas e minutos, numerais arábicos e romanos, palavras compostas, nomes próprios, caracteres especiais e de pontuação final. Os *tokens* são armazenados em estruturas de dados (*arrays*) e enviados um a um para o Identificador Morfológico, que é o módulo do Linguístico responsável pelo etiquetamento morfológico dos *tokens*, usando para isso, tecnologia adaptativa.

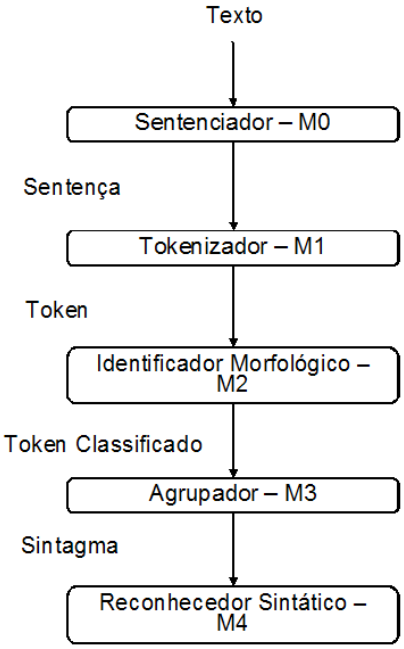

Figura 1. Estrutura do Linguístico.

O Identificador Morfológico é composto por um Autômato Mestre e um conjunto de submáquinas especialistas que acessam bases de dados de classificações morfológicas (Fig.2.1). A prioridade é obter as classificações de corpus já analisados e etiquetados; caso o termo procurado não seja encontrado, o Identificador Morfológico procura por verbos, substantivos e adjetivos, no formato finito e infinito (flexionados e não flexionados), através de uma submáquina de formação e identificação de palavras; por fim, o Identificador Morfológico procura por termos invariáveis, ou seja, termos cuja classificação morfológica é considerada estável pelos linguistas, tais como, conjunções, preposições e pronomes.

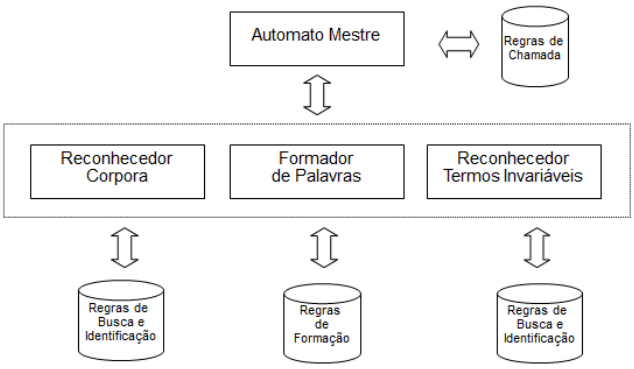

Figura 2.1. Arquitetura do Identificador Morfológico.

O Autômato Mestre (Fig.2.2) é responsável pelo sequenciamento das chamadas às submáquinas, de acordo com um conjunto de regras cadastradas em base de dados. Ele inicia o processamento recebendo o *token* da coleção de *tokens*. Em

seguida, o autômato se modifica através de uma função adaptativa, criando uma submáquina de processamento (M1), uma transição entre o estado 1 e M1, e uma transição que aguarda o estado final da submáquina M1. O token é passado para M1 e armazenado em uma pilha. Quando M1 chega ao estado final, o autômato se modifica novamente, criando uma nova submáquina M2, o estado 2 e as transições correspondentes. Caso o estado final de M1 seja de aceitação, o processo é finalizado no estado 2, caso contrário a máquina M2 é chamada, passando o token armazenado na pilha.

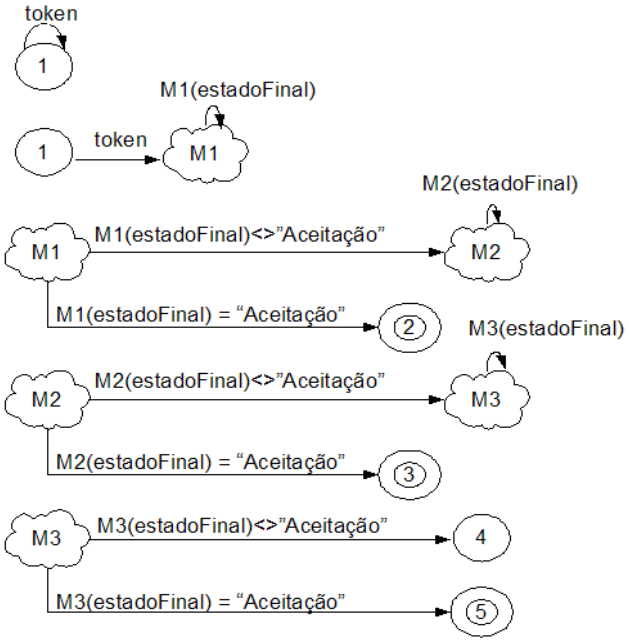

Figura 2.2. Estrutura Adaptativa do Autômato Mestre.

O processo se repete quando M2 chega ao final do processamento, com a criação da submáquina M3, do estado 3 e das transições correspondentes. Um novo ciclo se repete, e se o estado final de M3 é de aceitação, o autômato transiciona para o estado 5, de aceitação, caso contrário, ele vai para o estado 4 de não aceitação. M1, M2 e M3 representam, respectivamente, as submáquinas do Reconhecedor de Corpus, do Formador de Palavras e do Reconhecedor de Termos Invariáveis. Portanto, caso o Autômato Mestre encontre a classificação morfológica ao final de M1, ele não chama M2; caso encontre em M2, não chama M3 e, caso também não encontre em M3, ele informa aos demais módulos do Linguístico que não há classificação morfológica para o termo analisado.

As submáquinas M1, M2 e M3 também foram projetadas de acordo com a tecnologia adaptativa. A submáquina M1 usa um autômato adaptativo que se automodifica de acordo com o tipo de token que está sendo analisado: palavras simples, palavras compostas, números, valores e símbolos. A Fig. 2.3 apresenta a estrutura adaptativa de M1. Inicialmente o autômato é composto por um único estado e por transições para ele mesmo (Fig.2.3, à esquerda). Ao identificar o tipo de token que será analisado (obtido no processo de tokenização), o autômato cria submáquinas e as transições correspondentes.

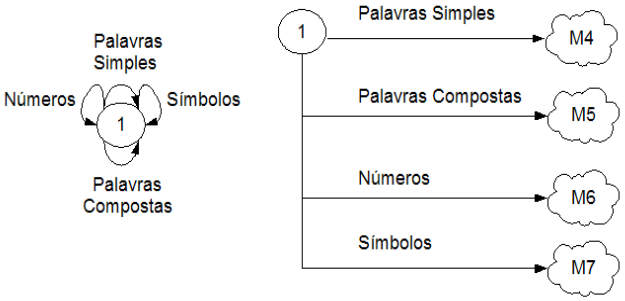

Figura 2.3. Estrutura Adaptativa do Reconhecedor de Corpus M1.

As alternativas de configuração são apresentadas na Fig.2.3, à direita. As submáquinas M4, M5, M6 e M7 reconhecem os tokens através de outro tipo de autômato que os processam sequencialmente de acordo com as letras, números e caracteres especiais que os compõem (Fig. 2.4).

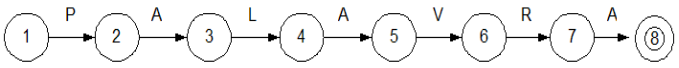

Figura 2.4. Reconhecedor de Palavras Simples.

No exemplo apresentado na Fig.2.4, o token "Palavra" é processado pela máquina M4; se o processamento terminar em um estado de aceitação, o token é reconhecido. As submáquinas M4, M5, M6 e M7 são criadas previamente por um programa que lê Corpus e os convertem em autômatos finitos determinísticos. A estrutura de identificação morfológica é composta por um par [chave, valor], no qual a chave é a classificação morfológica e o valor é o estado de aceitação do elemento lexical. No exemplo apresentado, a chave do item lexical "palavra" seria a classificação morfológica, N - substantivo e o estado "8" faria parte do conjunto de estados de aceitação, indicando que o *token* "palavra" seria sido aceito.

O Formador de Palavras, submáquina M2, também é montado previamente por um programa construtor, levando em consideração as regras de formação de palavras do Português do Brasil, descritas por Margarida Basílio em [13]. A submáquina M2 utiliza o vocabulário do TeP2.0[14], composto por verbos, substantivos e adjetivos, e o conjunto de regras de prefixação, sufixação e regressão verbal descrito pela autora para construir novas formas. No entanto, são necessários alguns cuidados para evitar a criação de estruturas que aceitem palavras inexistentes. A Fig. 2.5 apresenta um exemplo de autômato no qual são aplicados os prefixos "a" e "per", e os sufixos "ecer" e "ar" (derivação parassintética) ao radical "noit" do substantivo noite, que faz parte do TeP2.0.

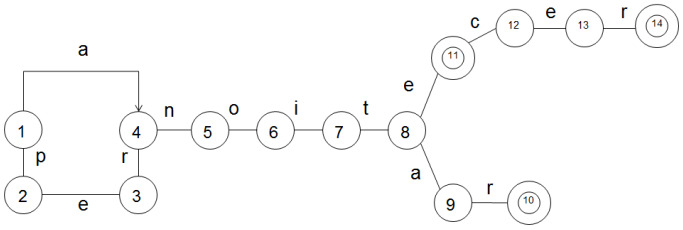

Figura 2.5. Autômato Formador de Palavras.

No exemplo apresentado, as palavras “anoitecer”, “pernoitar” e “anoitar” (sinônimo de “anoitecer”) existem no léxico do português do Brasil. Já pernoitecer é uma combinação que não existe na língua portuguesa. Margarida Basílio diz que algumas combinações não são aceitas simplesmente porque já existem outras construções consagradas pelo uso. Para reduzir o risco de aceitar derivações inexistentes, o processo de construção do autômato restringe as possíveis formações, utilizando apenas as regras que a autora destaca como sendo mais prováveis. É o caso de nominalização de verbos com o uso dos sufixos –ção , –mento e – da. A autora acrescenta que o sufixo –ção é responsável por 60% das formações regulares, enquanto o sufixo –mento é responsável por 20% destas formações. Já o sufixo –da é, via de regra, usado em nominalizações de verbos de movimento, tais como, entrada, saída, partida, vinda, etc.

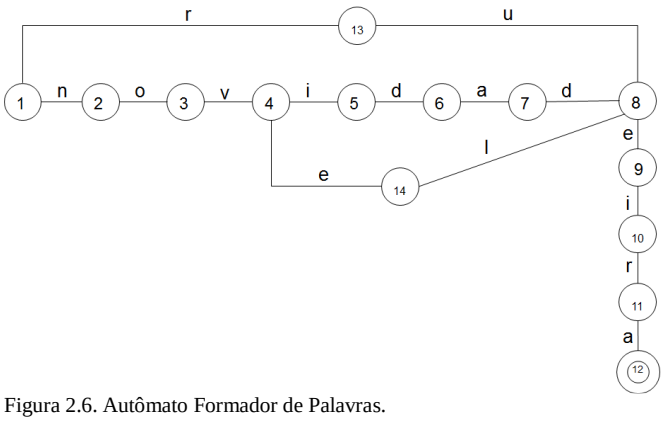

Figura 2.6. Autômato Formador de Palavras.

Outra característica do construtor do autômato é representar os prefixos e sufixos sempre pelo mesmo conjunto de estados e transições, evitando repetições que acarretariam o consumo desnecessário de recursos computacionais. A Fig. 2.6 apresenta exemplo de palavras formadas por reutilização de estados e transições na derivação das palavras rueira, novidadeira e noveleira. Os estados e transições usados para representar o sufixo “eira”, usado para designar são os mesmos nas três derivações.

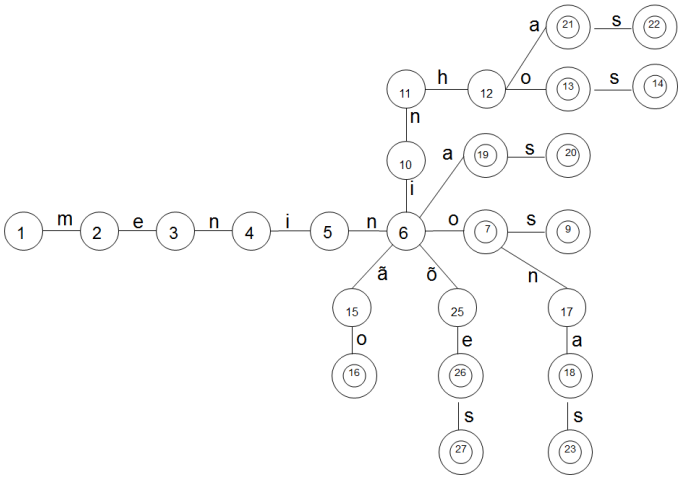

Figura 2.7. Autômato Formador de Flexões Nominais.

A submáquina M2 também reconhece palavras flexionadas, obtidas, no caso de substantivos e adjetivos, através da aplicação de sufixos indicativos de gênero, número e grau aos radicais do vocabulário TeP2.0. A Fig. 2.7 apresenta um exemplo de autômato usado para a formação das formas flexionadas do substantivo menino. Foram adicionados ao radical “menin” as flexões “o(s)”, “a(s)”, “ão”, “ões”, “onona(s)”, “inho(s)” e“inha(s)”.

No caso de verbos, foi criada uma estrutura de estados e transições para representar as flexões de tempo, modo, voz e pessoa, obtendo-se, assim, as respectivas conjugações.

A Fig. 2.8 apresenta um exemplo de autômato usado para a formação das formas flexionadas do presente do indicativo do verbo andar. Foram adicionados ao radical “and” as flexões “o”, “as”, “a”, “andamos”, “ais” e “andam”.

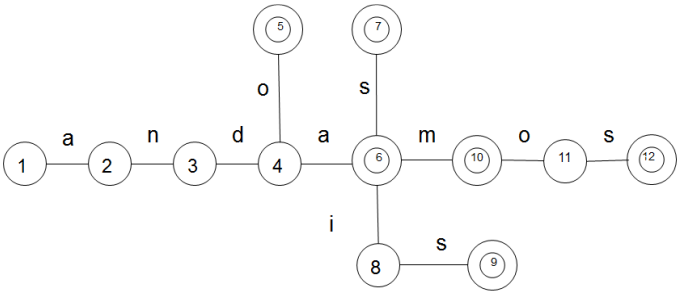

Figura 2.8. Autômato Formador de Flexões Verbais.

A estrutura de armazenamento da submáquina M2 também é composta por um par [chave, valor], no qual a chave é a classificação morfológica e o valor é o estado de aceitação do elemento lexical. Já a submáquina M3 é um autômato que varia em função do tipo de termo (conjunções, preposições e pronomes) e utiliza uma estrutura arbórea similar a M4. A Fig. 2.9 apresenta um exemplo de autômato usado no reconhecimento de termos deste domínio. A estrutura de armazenamento da submáquina M3 é composta por um par [chave, valor], no qual a chave é a classificação morfológica e o valor é o estado de aceitação do elemento lexical.

Figura 2.9. Autômato Reconhecedor de Conjunções, Preposições e Pronomes.

Como as palavras podem ter mais do que uma classificação morfológica, é necessário utilizar uma técnica para selecionar aquela que é mais apropriada para o contexto analisado. Por exemplo, a palavra “casa” pode ser classificada como substantivo comum, feminino, singular ou como verbo flexionado na 3ª pessoa do singular no tempo presente, modo indicativo. No entanto, tendo em vista o contexto em que palavra se encontra, é possível escolher a classificação mais provável. Por exemplo, se a palavra “casa” vier precedida de um artigo definido, é mais provável que ela seja um substantivo; já se a palavra antecessora for um substantivo próprio, é mais provável que “casa” seja um verbo.

O Automato Mestre utiliza o algoritmo Viterbi [19] para fazer as desambiguações. Este algoritmo encontra a sequência mais provável de etiquetas para uma determinada sequência de *tokens*. A Figura 2.10 apresenta um exemplo da aplicação do

algoritmo Viterbi para desambiguação do *token* "casa" na sequência "A casa". O algoritmo recebe como parâmetro de entrada o *token* que está sendo analisado e encontra a etiqueta mais provável, considerando a sequência de etiquetas anteriores e as possíveis etiquetas do *token* analisado.

		A	casa
Etiqueta	<I>	Artigo	?
Opção 1	<I>	Artigo	Substantivo
Opção 1	<I>	Artigo	P(Substantivo|Artigo,<I>)*E(casa|Substantivo)
Opção 2	<I>	Artigo	Verbo
Opção 2	<I>	Artigo	P(Verbo|Artigo,<I>)*E(casa|Verbo)
Resultado	<I>	Artigo	Substantivo - Maior probabilidade

Onde:

<I> = etiqueta que identifica início de frase
P = probabilidade da etiqueta avaliada, considerando as duas últimas
E = probabilidade do *token*, considerando a etiqueta avaliada

Figura 2.10. Exemplo de Aplicação do Algoritmo Viterbi.

II. EXPERIMENTOS E RESULTADOS COMPARATIVOS

O Identificador Morfológico foi desenvolvido em linguagem Python, com o apoio do NLTK [16], uma biblioteca de processamento de linguagem natural. As submáquinas M1, M2 e M3 foram criadas a partir do vocabulário TeP2.0. No caso de verbos, foram implementados estados e transições para representar as flexões de tempo, modo, voz e pessoa, obtendo-se, assim, as respectivas conjugações. No caso dos substantivos e dos adjetivos, foram implementados estados e transições para representar as flexões de número.

O algoritmo Viterbi foi implementado através de um mecanismo de *fall-back*, iniciando a análise com trigramas, passando por bigramas e unigramas, e assumindo a classificação mais comum do corpus, que é substantivo, no caso de não encontrar nenhuma classificação. O mecanismo foi inicialmente treinado com 70% do corpus CINTIL Treebank [17], ficando os 30% restantes para testes. O tamanho do corpus de treino foi, então, progressivamente ampliado, para 75%, 80%, 85% e 90% do corpus CINTIL, deixando sempre o restante para testes. Os resultados foram apurados através do indicador de acurácia (Total de etiquetas corretas/Total de etiquetas *gold standard*) e os resultados são apresentados na Tab. 2.

Corpus	70%	75%	80%	85%	90%
Acurácia	90,69%	90,73%	90,93%	91,02%	91,41%

Tabela 2. Resultados do Etiquetador sem Tecnologia Adaptativa

Em seguida, as submáquinas M1, M2 e M3 foram incorporadas ao mecanismo de *fall-back* de forma a ser chamadas após o processamento dos unigramas e antes do mecanismo de *fall-back* assumir a classificação mais comum do corpus. A ideia era testar se as submáquinas melhoravam o desempenho do Identificador Morfológico e qual o ganho obtido, caso isso ocorresse.

Corpus	70%	75%	80%	85%	90%
Acurácia	91,98%	92,01%	92,16%	92,31%	92,49%
Diferença	1,29%	1,28%	1,23%	1,29%	1,08%

Tabela 3. Resultados do Etiquetador com Tecnologia Adaptativa

Os resultados apresentados na Tab. 3 mostraram que o uso da tecnologia adaptativa proporcionou ganhos consistentes em todos os cenários de testes, sendo maiores quando o tamanho do corpus de treinamento foi menor. Como as submáquinas ainda não incorporaram todas as regras linguísticas previstas no modelo, e o tamanho do vocabulário usado para montagem dos autômatos foi relativamente pequeno, é possível que o desempenho do Identificador Morfológico ainda possa ser melhorado, com o objetivo de chegar ao nível dos etiquetadores do estado da arte da Língua Portuguesa que chegam a 97% de acurácia [18,19].

Também foi testado o cenário no qual o tamanho do corpus de treino é relativamente pequeno, e se a tecnologia adaptativa poderia trazer melhores resultados neste cenário. Para isso, foram realizados dois outros experimentos. No primeiro, o tamanho do corpus de treinamento foi treinado inicialmente com 40% do corpus CINTIL e testado com o restante. Em seguida, o tamanho do corpus de treinamento foi reduzido para 35%, 30%, 25% e 5%, ficando sempre o restante para testes. Os resultados são apresentados na Tab.4.

Corpus	40%	35%	30%	25%	5%
Acurácia	88,66%	88,04%	87,33%	86,35%	64,90%

Tabela 4. Resultados do Etiquetador sem Tecnologia Adaptativa

No segundo experimento, os testes foram repetidos com o uso das submáquinas adaptativas. Os resultados são apresentados na Tab. 5.

Corpus	40%	35%	30%	25%	5%
Acurácia	90,47%	89,99%	89,35%	88,55%	67,06%
Diferença	1,81%	1,95%	2,02%	2,20%	2,16%

Tabela 5. Resultados do Etiquetador com Tecnologia Adaptativa

Percebe-se, novamente, que o uso da tecnologia adaptativa melhorou o desempenho do etiquetamento, neste caso, gerando resultados melhores do que no cenário anterior, o que permite concluir que quando o tamanho do corpus é reduzido, a influência da tecnologia adaptativa é mais significativa. Um aspecto interessante é imaginar o que aconteceria se fossem utilizadas apenas a tecnologia adaptativa e as regras linguísticas, pois esta configuração poderia ser interessante no cenário em que não houvesse corpus de treinamento ou se o tamanho do corpus treinamento fosse insuficiente para o uso de técnicas estatísticas.

III. CONSIDERAÇÕES FINAIS

Este artigo apresentou uma revisão dos conceitos de Tecnologia Adaptativa e de Processamento da Linguagem Natural, e, em seguida, descreveu a arquitetura do reconhecedor gramatical Linguístico, com ênfase no módulo Identificador Morfológico. Em seguida, foram apresentados experimentos nos quais as submáquinas adaptativas de etiquetamento morfológico foram avaliadas, comparando seu desempenho ao do etiquetamento realizado sem o uso de tecnologia adaptativa. Os resultados permitiram concluir ganho consistente com o uso de tecnologia adaptativa e possibilidade de obtenção de melhorias, visto que nem todas as regras linguísticas foram implementadas nas submáquinas usadas nos experimentos. Os experimentos também geraram um desdobramento, no sentido

de avaliar o desempenho do Identificador Morfológico, usando apenas tecnologia adaptativa e regras linguísticas, configuração que pode ser útil quando não houver corpus de treinamento ou quando o tamanho do corpus treinamento for insuficiente para o uso de técnicas estatísticas.

REFERÊNCIAS

[1] http://www.pcs.usp.br/~lta/

[2] Taniwaki, C. Formalismos adaptativos na análise sintática de Linguagem Natural. Dissertação de Mestrado, EPUSP, São Paulo, 2001.

[3] Menezes, C. E. Um método para a construção de analisadores morfológicos, aplicado à língua portuguesa, baseado em autômatos adaptativos. Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, 2000.

[4] Padovani, D. Uma proposta de autômato adaptativo para reconhecimento de anáforas pronominais segundo algoritmo de Mitkov. Workshop de Tecnologias Adaptativas – WTA 2009, 2009.

[5] Moraes, M. Alguns aspectos de tratamento sintático de dependência de contexto em linguagem natural empregando tecnologia adaptativa, Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, 2006.

[6] Rich, E.; Knight, K. Inteligência Artificial, 2. Ed. São Paulo: Makron Books, 1993.

[7] Vieira, R.; Lima, V. Linguística computacional: princípios e aplicações. IX Escola de Informática da SBC-Sul, 2001.

[8] Fuchs, C.; Le Goffic, P. Les Linguistiques Contemporaines.

[9] M. G. V.Nunes et al. Introdução ao Processamento das Línguas Naturais. Notas didáticas do ICMC Nº 38, São Carlos, 88p, 1999. Paris, Hachette, 1992. 158p.

[10] Sardinha, T. B. A Língua Portuguesa no Computador. 295p. Mercado de Letras, 2005.

[11] Rocha, R.L.A. Tecnologia Adaptativa Aplicada ao Processamento Computacional de Língua Natural. Workshop de Tecnologias Adaptativas – WTA 2007, 2007.

[12] Contier, A., Padovani D., Neto J.J. O reconhecedor gramatical adaptativo Linguístico: experimentos e resultados comparativos. Workshop de Tecnologia Adaptativa (11: 2017: São Paulo) Memórias do WTA 2017. – São Paulo; EPUSP, 2017. 138.

[13] Basilio, M. Formação e Classes de Palavras no Português do Brasil. Ed.Contexto, 2004.

[14] http://www.nilc.icmc.usp.br/tep2/

[15] Forney, G.D. J. IEEE. Proceedings of the IEEE, Volume 61, Issue 3, 1973.

[16] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.

[17] Branco, António, Francisco Costa, João Silva, Sara Silveira, Sérgio Castro, Mariana Avelãs, Clara Pinto and João Graça, 2010, "Developing a Deep Linguistic Databank Supporting a Collection of Treebanks: the CINTIL DeepGramBank ", In Proceedings, LREC2010 - The 7th international conference on Language Resources and Evaluation, La Valleta, Malta, May 19-21, 2010.

[18] Branco, A., Silva, J.: Evaluating solutions for the rapid development of state-of-the-art POS taggers for Portuguese. In: Proceedings of the 4th Language Resources and Evaluation Conference (LREC). (2004) 507–510.

[19] Silva, J.: Shallow processing of Portuguese: From sentence chunking to nominal lemmatization. Master's thesis, University of Lisbon (2007) Published as Technical Report DI-FCUL-TR-07-16.

Ana Teresa Contier formou-se em Letras-Português pela Universidade de São Paulo (2001) e em publicidade pela PUC-SP (2002). Em 2007 obteve o título de mestre pela Poli-USP com a dissertação: "Um modelo de extração de propriedades de textos usando pensamento narrativo e paradigmático".

Djalma Padovani formou-se em administração de empresas pela Faculdade de Economia e Administração da Universidade de São Paulo, em 1987 e obteve o mestrado em engenharia de software pelo Instituto de Pesquisas Tecnológicas de São Paulo - IPT, em 2008. Trabalhou em diversas empresas nas áreas de desenvolvimento de software e tecnologia de informação e atualmente atua no Laboratório de Dados da Serasa Experían.

João José Neto graduado em Engenharia de Eletricidade (1971), mestrado em Engenharia Elétrica (1975) e doutorado em Engenharia Elétrica (1980), e livre-docência (1993) pela Escola Politécnica da Universidade de São Paulo. Atualmente é professor associado da Escola Politécnica da Universidade de São Paulo, e coordena o LTA - Laboratório de Linguagens e Tecnologia Adaptativa do PCS - Departamento de Engenharia de Computação e Sistemas Digitais da EPUSP. Tem experiência na área de Ciência da Computação, com ênfase nos Fundamentos da Engenharia da Computação, atuando principalmente nos seguintes temas: dispositivos adaptativos, tecnologia adaptativa, autômatos adaptativos, e em suas aplicações à Engenharia de Computação, particularmente em sistemas de tomada de decisão adaptativa, análise e processamento de linguagens naturais, construção de compiladores, robótica, ensino assistido por computador, modelagem de sistemas inteligentes, processos de aprendizagem automática e inferências baseadas em tecnologia adaptativa.