Contribuições à Modelagem Adaptativa da Norma Culta do Português Brasileiro

D. Padovani e J. J. Neto

***Resumo*— Este trabalho apresenta conceitos de processamento de língua natural, abordando diferentes métodos e discorrendo sobre os principais problemas encontrados para a realização desta tarefa. Em seguida, é feita uma revisão dos trabalhos desenvolvidos no tema, com enfoque no processamento do Português do Brasil. Por fim, é apresentada uma proposta de modelo unificado que usa o formalismo adaptativo como modelo teórico subjacente, sem a necessidade de recorrer a técnicas auxiliares.**

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

Processamento da Linguagem Natural

A língua natural é o meio pelo qual os seres humanos se comunicam. Ela é caracterizada pela riqueza semântica, léxica e sintática, que permite a elaboração de textos complexos e com alto grau de abstração, tais como os vistos nas grandes obras da literatura, ou precisos e direcionados, como aqueles encontrados em tratados, trabalhos acadêmicos e científicos. A língua natural não permanece estanque, ao contrário, está sempre em permanente evolução, agregando novos termos e estruturas e eliminando outros, em um processo constante de adaptação às necessidades impostas pela realidade.

Há um grande apelo do ponto de vista dos seres humanos em se comunicar com uma máquina da mesma forma que o fazem entre si. Muitas pessoas encontram dificuldades para utilizar os dispositivos convencionais de interação com os computadores, que, em maior ou menor grau, restringem as possibilidades linguísticas para que as instruções possam ser interpretadas e convertidas em comandos que a máquina possa executar. Estas limitações podem gerar desconforto e, em alguns casos, rejeição; daí a grande procura de computadores e sistemas que interpretem a língua natural. Datam da década de 1940 as primeiras pesquisas que visavam possibilitar o uso da língua natural como forma de interação entre o homem e a máquina [1]. Existem trabalhos que estudam a geração de programas a partir de textos [2], elaboração de tradutores, revisores e corretores gramaticais [3],[4], identificação de padrões e extração de resumos [5], geração automática de padrões de extração [6] e geração automática de taxonomias hierárquicas [7].

O processamento da língua natural requer o desenvolvimento de programas que sejam capazes de determinar e interpretar a estrutura léxico-sintática e semântica das sentenças em muitos níveis de detalhe. As línguas naturais exibem um intricado comportamento estrutural visto que são profusos os casos particulares a serem considerados, devido à extrema generalização ou especialização adotadas nas modelagens usuais das regras gramaticais vigentes na literatura. Uma vez que as línguas naturais nunca são formalmente projetadas, suas regras sintáticas não são simples nem tampouco óbvias e tornam, portanto, complexo o seu processamento computacional. Diversas abordagens são empregadas em sistemas de processamento de língua natural, tais como os métodos exatos, aproximados, pré-definidos ou interativos, inteligentes ou algorítmicos [8]. Independentemente do método utilizado, o processamento da língua natural envolve as operações de análise léxico-morfológica, análise sintática, análise semântica e análise pragmática [9].

Os principais problemas enfrentados no processamento da língua natural estão relacionados ao tratamento das ambiguidades e aos não-determinismos. Rocha [10] aponta que os primeiros resultados animadores no processamento de língua atual foram resultados de técnicas estatísticas. No entanto, ressalta, nenhum modelo estatístico conseguiu resolver os problemas mais complexos de processamento da língua natural e que o uso indiscriminado de modelos estatísticos apenas comprovou a necessidade de evitar uma resposta única. Entre as propostas atuais, o autor relaciona 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 ou o uso de modelos estatístico como base para escolha de modelos não determinísticos.

Nota-se que não há um consenso sobre qual método usar e também não se encontram estudos a respeito dos contextos em que os métodos são mais adequados, nem tampouco, sobre quando são mais eficientes e como transitar de um método para outro. Visto que nenhuma técnica isoladamente resolve completamente o problema do processamento da língua natural, aparentemente existe uma lacuna na literatura relacionada ao tema.

Revisão da Literatura

Ladeira [11] faz um levantamento produção científica nacional realizada no campo de processamento da língua natural nos últimos 40 anos, categorizando-a e analisando a evolução dos grandes temas neste período. Dentre as 621 publicações consideradas da área, a autora definiu um material empírico, constituído por uma amostra de 68 trabalhos, que foi submetido à análise de conteúdo, que tinha por objetivo elucidar as temáticas discutidas pela comunidade científica nacional no campo do processamento da língua natural. A autora conclui que a maior parte da produção científica nacional foi publicada depois do ano 2000. Poucos grupos foram responsáveis por grande parte da produção nacional, sendo a maioria proveniente da ciência da computação, linguística e engenharia elétrica. Com relação à evolução das problemáticas mais discutidas, o trabalho de tradução foi intensamente abordado na década de 70; os estudos com indexação diminuíram a partir da década de 80; e as pesquisas sobre classificação passaram por um período de dormência na

década de 90, notando-se uma clara tendência no desenvolvimento de pesquisas em sumarização automática. A análise de conteúdo realizada nas 68 publicações selecionadas revelou que a recuperação de informação foi a problemática que teve maior destaque na produção científica nacional com 18 trabalhos, seguida da temática de sumarização (10 trabalhos), tratamento de ambiguidade e parsers (10 trabalhos), tradução (9 trabalhos), aplicações para a própria área e exemplos de aplicações de processamento de língua natural (4 trabalhos) e correção automática (3 trabalhos). Dos trabalhos analisados sobre sumarização, observou-se que a maioria das pesquisas tem privilegiado a abordagem empírica para gerar extratos. As pesquisas em tradução automática têm utilizado métodos estatísticos e regras de transferências, com resultados muito próximos.

Especificamente, com relação à problemática de *parsers*, a autora identificou três níveis de análise da língua natural: análise léxico-morfológica, análise sintática e análise semântica. Além disso, ela observou que alguns trabalhos abordaram dois ou até mesmo três níveis de análise. Dentre os trabalhos que propõem análise léxico-morfológica, Neto e Menezes [12] apresentam um método para a construção de um etiquetador morfológico, que pode ser usado em várias línguas. Segundo os autores, existem, basicamente, quatro métodos de etiquetagem morfológica de textos em língua natural: o estatístico, o que se utiliza de regras escritas manualmente, o baseado em regras inferidas automaticamente; e o com base em exemplos memorizados. Os autores acrescentam que todos os métodos utilizam três fontes de informação linguística, 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 a categorias morfológicas (léxico), para fornecer informações sobre palavras conhecidas; e o contexto próximo ao item lexical que se quer etiquetar (2 ou 3 etiquetas ao redor), para refinar a escolha de sua etiqueta. Assim, o método proposto etiqueta primeiro as palavras conhecidas, depois as desconhecidas usando heurística de acordo com o sufixo, e finalmente faz um refinamento, de acordo com o contexto.

Padilha e Viccari [13] propõem o desenvolvimento de processadores para a morfologia do português utilizando máquinas de estados finitos (transdutores). Os autores alegam que os transdutores são adequados para o processamento morfológico da língua portuguesa, mas ressaltam, como limitações, a sua construção generativa (não há algoritmos de aprendizado de novas transformações, nestes casos, a gramática deve ser alterada e o transdutor reconstruído); e a ausência de pesos para diferenciar mapeamentos ambíguos.

Dentre os trabalhos que abordaram a análise sintática está o trabalho de Bonfante e Nunes [14] que propõem um *parser* probabilístico baseado na noção de núcleos lexicais, na qual, para cada regra observada no conjunto de treinamento, as palavras que não são núcleo são chamadas de modificadores, exercendo influência sobre ele. Segundo as autoras a grande dificuldade de se especificar uma gramática com poder de descrição abriu caminho para a pesquisa empírica. Assim, um conjunto de sentenças anotadas sintaticamente é usado, como dados de treinamento, num processo de aprendizado para realizar a anotação de uma sentença desconhecida. Dentre as abordagens empíricas, as autoras citam o aprendizado de máquina simbólico, conexionista e estatístico. A formação da estrutura sintática de uma sentença se dá através de um processo *bottom-up* comandado pela probabilidade de um núcleo e um modificador se juntarem para formar um sintagma. Utilizou-se um conjunto de sentenças obtidas do corpus NILC e anotadas sintaticamente pelo *parser* PALAVRAS [15].

Julia, Seabra e Semeghini-Siqueira [16] propõem um *parser* que realiza a análise sintática e semântica de afirmações sobre especificação de software expressas de maneira irrestrita em língua natural. O analisador proposto corresponde a uma estrutura, como definido por Piaget [17], que automaticamente gera regras semânticas durante a análise, através de um método heurístico. Segundo os autores, uma estrutura é um sistema de transformações caracterizadas por um grupo de regras. A parte sintática da gramática é expressa por meio de regras, tais como as regras de gramática proposta por Chomsky [18]. O *parser* é baseado em algoritmos de busca que têm como objetivo encontrar um caminho da árvore sintática até um nó folha que contenha uma categoria de significado. A categoria de cada palavra na sentença irá depender da ordem em que ela aparece na sentença.

Sardinha [19] apresenta pesquisas realizadas na área de processamento de língua natural para a Língua Portuguesa, procurando criar um panorama da produção científica não apenas proveniente do Brasil, mas também de Portugal, França e Dinamarca. Nunes et al. [20] apresentam, entre outras ferramentas, o *parser* Curupira, desenvolvido pelo NILC. O Curupira é um analisador de propósito geral para Português do Brasil. Ele analisa sentenças de cima para baixo e da esquerda para a direita através de uma gramática funcional livre de contexto para o padrão escrito Português do Brasil e um léxico de ampla cobertura para Português do Brasil. Este último é um conjunto de 1,5 milhões de formas livres (incluindo formas flexionadas e derivadas), com informações morfossintáticas, tais como *part-of-speech*, número, pessoa, gênero, tempo, aspecto, e transitividade. O Curupira não faz uso de estratégias de poda, redução ou simplificação, procurando etiquetar cada um dos itens lexicais da sentença, por menores ou menos expressivos que sejam. O Curupira também não desambigua a estrutura sintática das sentenças da língua portuguesa, fornecendo todas as classificações sintáticas segundo as prioridades das regras gramaticais que lhe servem de base. O Curupira parte do pressuposto de que as sentenças informadas estão corretas, informando todas as possibilidades sintáticas de acordo a gramática subjacente. O Curupira também não realiza análise semântica e, como trabalha combinando palavras e classes de palavras para fornecer as possíveis estruturas sintáticas, pode gerar classificações que não fazem sentido no contexto.

Bick [15] apresenta pesquisa desenvolvida pelo projeto VISL – *Visual Interactive Syntax Learning*, sediado na Universidade do Sul da Dinamarca, na qual também usa a abordagem determinística no desenvolvimento do *parser* PALAVRAS, um analisador reducional que seleciona as etiquetas com base em regras de constrição da Gramática Constritiva proposta por Karlson [21]. Sardinha [19] explica que esta não é uma gramática tradicional, que explica as categorias gramaticais e sintáticas, e, sim, um conjunto de

regras formalizadas de tal modo que possam ser usadas por computadores. O PALAVRAS procura desambiguar possíveis interpretações morfológicas através da aplicação de regras que utilizam condições contextuais para restringir as possíveis classificações, selecionando, ao final, a etiqueta mais adequada. Bick explica que, no nível sintático, o *parser* trabalha com regras produtivas e restritivas, sendo que as primeiras mapeiam etiquetas ambíguas e as últimas rejeitam as etiquetas com base no contexto.

Uma abordagem estatística é apresentada por Marques e Lopes [22]. Os autores partem do pressuposto de que para realizar a tarefa de etiquetagem morfossintática é necessário um grande volume de texto anotado manualmente, com centenas de milhares de palavras desambiguadas morfossintaticamente e dizem não existir corpora com as dimensões necessárias na língua portuguesa, nem tampouco disponibilidade para a construção de um corpus com estas características. Os autores também refutam a construção de um sistema baseado em regras, pois estes também requerem a interferência de uma pessoa com conhecimento para fornecer regras tão específicas e dependentes do texto que o sistema vai processar. A alternativa que propõem é o uso de uma rede neuronal combinada com um sistema de análise lexical e um texto manualmente etiquetado com 5400 palavras. Segundo os autores a precisão de etiquetagem obtida por este modelo ultrapassa ligeiramente 98%. Outra abordagem estatística é empregada por Dias e Lopes [23], na extração automática de unidades poli lexicais para o português. Os autores defendem que o uso de regras sintáticas para extrair unidades poli lexicais contíguas requer um conhecimento muito profundo da língua, tonando a abordagem pouco flexível para aplicação a novas línguas, assim como pouco adequada ao caráter evolutivo e dinâmico da língua.

Neto [24] apresenta outro tipo de abordagem com a introdução do conceito de adaptatividade. Segundo o autor, dispositivos adaptativos são abstrações matemáticas capazes de se modificarem dinamicamente, criando formalismos capazes de se auto modificarem autonomamente. São compostos de um conjunto de regras, denominado dispositivo subjacente, tipicamente não adaptativo, e de um mecanismo adaptativo cuja conexão ao dispositivo subjacente proporciona-lhe os recursos complementares necessários para a realização de tarefas responsáveis pela auto modificação autônoma que os caracteriza. O mecanismo adaptativo é composto de dois elementos: as declarações das funções adaptativas e as associações das funções adaptativas ao dispositivo subjacente, denominadas ações adaptativas. Cada regra pode estar associada a duas ações adaptativas: uma antes e outra após a execução da regra. Se não houver nenhuma ação adaptativa a uma determinada regra, ele se comporta como uma regra não adaptativa. Entre os principais campos de aplicação da Tecnologia Adaptativa encontram-se as Linguagens de Programação, o Processamento de Línguas Naturais, a Computação Natural, a Inteligência Artificial e a Engenharia de Software, Reconhecimento de Padrões, Tomada de Decisão Robótica e Aprendizagem de Máquina. Dispositivos adaptativos apresentam forte potencial de aplicação para a construção de um modelo computacional unificado, pois permitem representar fenômenos linguísticos complexos, tais como dependências de contexto, além de serem usados como um formalismo de reconhecimento, o que permite seu uso no pré-processamento de textos para análise morfossintática, verificação de sintaxe, processamento para traduções automáticas, interpretação de texto e correção gramatical.

Observa-se que na literatura analisada não há referência a um modelo computacional unificado que possa representar as diferentes abordagens usadas no processamento da linguagem natural: determinística, estatística e heurística. Os trabalhos analisados não se beneficiam de técnicas diferentes das que seu modelo utiliza, perdendo com isso, a possibilidade de melhorar os resultados obtidos, ou mesmo, de buscar o aprimoramento do processo como um todo. O processamento puramente estatístico pode levar a necessidade de análise de cadeias que não são permitidas pelas regras gramaticais vigentes e que, no entanto, acabam fazendo parte do contexto da análise em virtude de representarem uma possível combinação estatística. Por outro lado, um modelo puramente determinístico requer a identificação exaustiva de todas as regras utilizadas na formalização da língua, o que além de requerer conhecimento muito profundo da língua, nem sempre disponível, torna a abordagem pouco adequada ao caráter evolutivo e dinâmico da língua. Heurísticas, por outro, lado, não oferecem embasamento teórico para serem utilizadas como abordagem principal no processamento da língua natural, apresentando-se muito mais como instrumento de apoio para resolução de ambiguidades.

Nota-se que os trabalhos refletem as limitações de cada modelo, por exemplo, o *parser* Curupira não desambigua a estrutura sintática das sentenças da língua portuguesa, o que talvez pudesse fazer, caso também utilizasse modelos estatísticos. O *parser* PALAVRAS utiliza milhares de regras da gramática constritiva, o que certamente lhe confere precisão, porém também o limita para acompanhar a evolução da língua, que requer sempre novas regras de representação. Por outro lado, no enfoque estatístico usado no etiquetador morfológico de Marques e Lopes [22], a etiqueta escolhida é a que apresenta maior probabilidade em função da sequencia de palavras e sequencia de etiquetas observadas no corpus. Este método desconsidera etiquetas válidas, porém com baixa probabilidade de ocorrência. A abordagem proposta por Neto [24] aparentemente é mais robusta, pois permite incorporar elementos determinísticos e probabilísticos no mesmo modelo, que se adapta em função do contexto observado, adicionando ou eliminando funções adaptativas. Além disso, em um mesmo modelo é possível representar características de linguagens dependentes de contexto, livres de contexto e irregulares, o que o torna, além de consistente, flexível para a construção de parsers e reconhecedores gramaticais.

PROPOSTA DE MODELO UNIFICADO

Em [25], os autores apresentam o Linguístico, um reconhecedor gramatical que usa técnicas adaptativas para reconhecimento da estrutura morfossintática de textos em Português Brasileiro, incorporando técnicas probabilísticas para desambiguação morfológica. A estrutura do Linguístico é apresentada na Fig. 1. A máquina M0 – Sentenciador é responsável pela primeira etapa do processamento, dividindo o texto inicial nas sentenças que o compõem. Em seguida, as sentenças são divididas em *tokens* por meio da máquina M1 –

Tokenizador, que identifica palavras e caracteres de pontuação. No passo seguinte, a máquina M2 – Identificador Morfológico classifica os *tokens* morfologicamente, com o auxilio de textos de apoio, tais como o Bosque [26] e o Tep 2.0 [27] e também com o uso de autômatos que desenvolvem conjugações e montam palavras a partir de regras de formação da Língua Portuguesa.

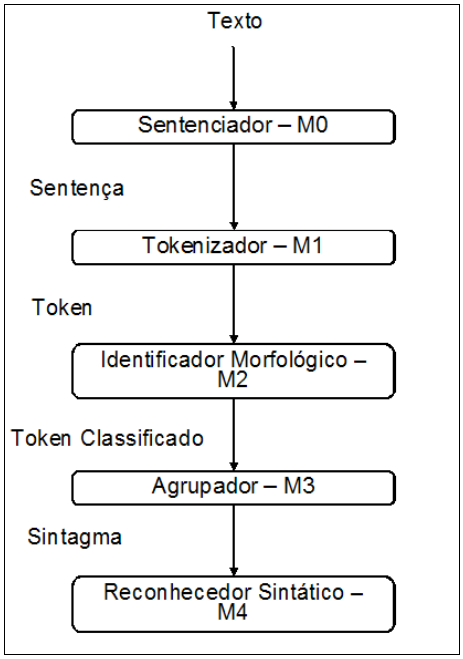


Figura 1. Estrutura do Linguístico [25]

Com os *tokens* classificados, a máquina M3 - Agrupador realiza a montagem de sintagmas, utilizando, para isso, um autômato, uma tabela de agrupamento, e as regras definidas na gramática de Celso Luft [28]. Por fim, os sintagmas são enviados para a máquina M4 – Reconhecedor Sintático, que os analisa e verifica se eles compõem sentenças válidas, segundo as regras sintáticas da gramática de Luft. Embora o Linguístico utilize técnicas determinísticas e probabilísticas, elas são apresentadas de forma pontual, para resolver partes específicas do processamento da língua, não sendo mencionado em nenhum momento, um modelo unificado subjacente que possa ser generalizado para qualquer situação.

A proposta apresentada neste artigo é uma evolução do Linguístico para as tarefas de reconhecimento e desambiguação morfológica e sintática. Como premissa, pressupõe-se que os *tokens* já estejam identificados antes do início da análise morfológica, o que corresponde às duas primeiras etapas do processamento realizado pelo Linguístico. A mudança inicia-se com a máquina M2 – Identificador Morfológico, que seria alterada para ser um gerador de classificações morfológicas e respectivas probabilidades de ocorrências. As máquinas M3 e M4 seriam incorporadas no mesmo autômato adaptativo, consolidando toda a tarefa de reconhecimento e desambiguação em um único modelo computacional, sem necessidade de tabelas de agrupamento ou outro recurso de apoio. A Fig.2 apresenta a nova estrutura do Linguístico modificado de acordo com a proposta.

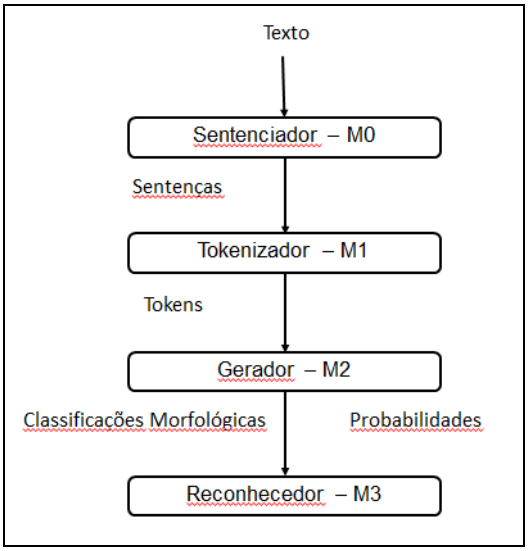


Figura 2. Estrutura Modificada do Linguístico

A máquina M3 – Reconhecedor é composta por um autômato que possui inicialmente um estado, um vetor de pilhas e uma função de transição adaptativa, responsável por criar os passos seguintes. Como o modelo incorpora técnicas probabilísticas, a cada passo a função adaptativa cria dois estados, um para representar a probabilidade de ocorrência do trigrama formado pelas três últimas classificações morfológicas e outro para representar a probabilidade do *token* analisado ser da classificação morfológica indicada pelo último símbolo do trigrama. Sempre que o autômato chega a um estado em que uma classificação morfológica é encontrada, ele atualiza o vetor de pilhas para controlar a montagem de sintagmas e fazer o reconhecimento sintático.

A Fig.3 apresenta a configuração inicial do autômato e vetor de pilhas. A função adaptativa (α(i)) é usada para criar dinamicamente os estados e transições conforme os *tokens* são processados. No estado inicial, não há indicação de movimentação na fila e ela encontra-se vazia. Usando como exemplo a frase “A casa estava aberta”, pode-se acompanhar o mecanismo de reconhecimento e desambiguação.

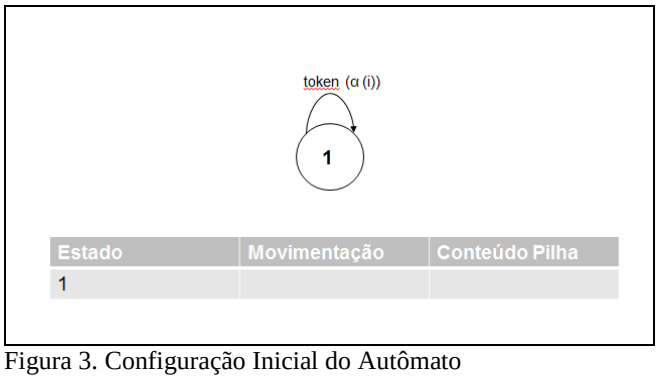
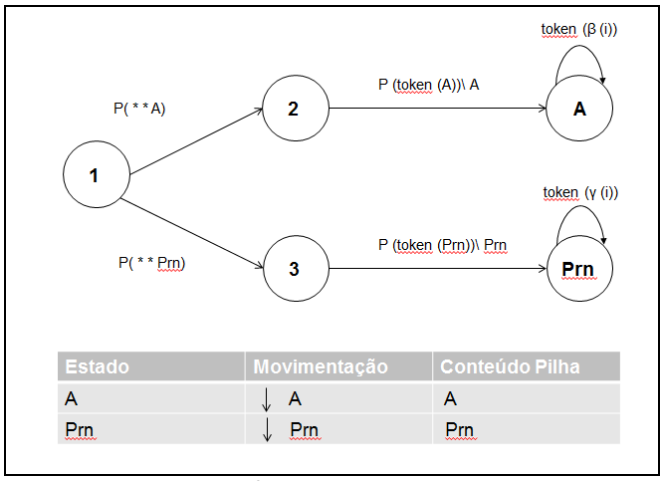


Figura 3. Configuração Inicial do Autômato

A Fig.4 apresenta a configuração do autômato antes processar o *token* “A”. Como existem duas classificações possíveis (artigo e pronome), a função adaptativa (α(i)) cria duas sequências de estados com suas respectivas transições, sendo uma para indicar a probabilidade de “A” ser artigo e outra para indicar a probabilidade de “A” ser pronome. O

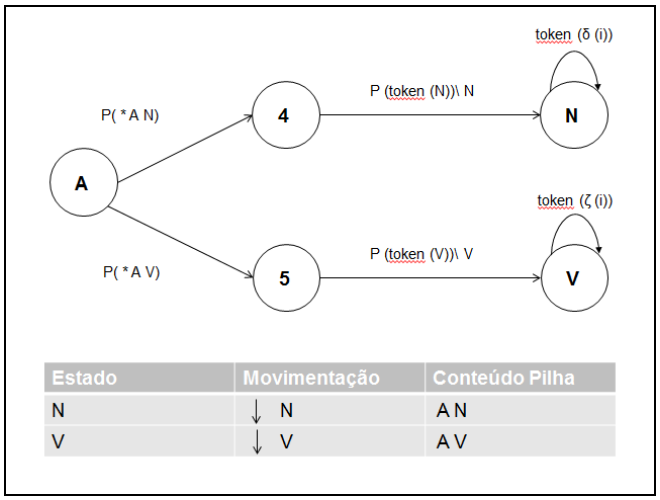
estado 2 representa o trigrama formado pela sequência * * A, na qual o * * representam as classificações morfológicas antes do inicio da frase e A indica artigo. O estado A indica o estado no qual foi identificado um artigo. Entre o estado 1 e o estado 2 indica-se a probabilidade de ocorrência do trigrama * * A e, entre o estado 2 e o estado A, a probabilidade do *token* "A" ser um artigo dado que os dois últimos *tokens* são * *. Ao chegar ao estado A, insere-se um símbolo "A" na pilha (representado por ⧫ A) para indicar que um *token* "A" foi encontrado. O processo ocorre de maneira análoga entre os estados 1, 3 e Prn. Ao final, (α(i)) cria outras duas funções adaptativas (β (i)) e (γ (i)) para prosseguir com a análise.



Estado	Movimentação	Conteúdo Pilha
A	↓ A	A
Prn	↓ Prn	Prn

Figura 4. Primeira transformação adaptativa

A Fig. 5 representa o passo seguinte. Neste caso, o *token* analisado é "casa", que pode ser classificado como substantivo ou verbo. Como o passo anterior criou dois estados, o reconhecimento continua a partir deles. Em A, a função adaptativa (β (i)) cria os estados 4, 5, N e V, sendo que N e V representam substantivos e verbos, respectivamente.

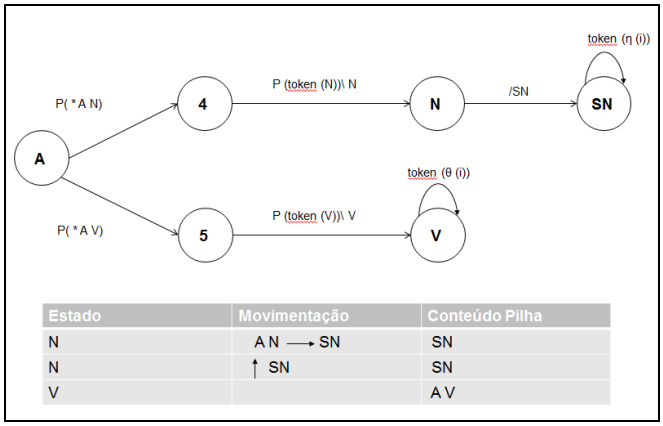


Estado	Movimentação	Conteúdo Pilha
N	↓ N	A N
V	↓ V	A V

Figura 5. Segunda transformação adaptativa

A transição entre o estado A e o estado 4 indica a probabilidade de ocorrência do trigrama * A N e a transição do estado 4 para o estado N indica a probabilidade do *token* "casa" ser um substantivo. Ao final, um símbolo N é armazenado na pilha e a função adaptativa (δ (i)) é criada para prosseguir com o reconhecimento a partir do estado N. O processo é análogo entre os estados A, 5 e V, sendo que, ao final, cria-se uma nova pilha com o símbolo inicial A e armazena-se o símbolo V. A função adaptativa (ζ (i)) é criada no estado V para prosseguir com o reconhecimento a partir deste estado.

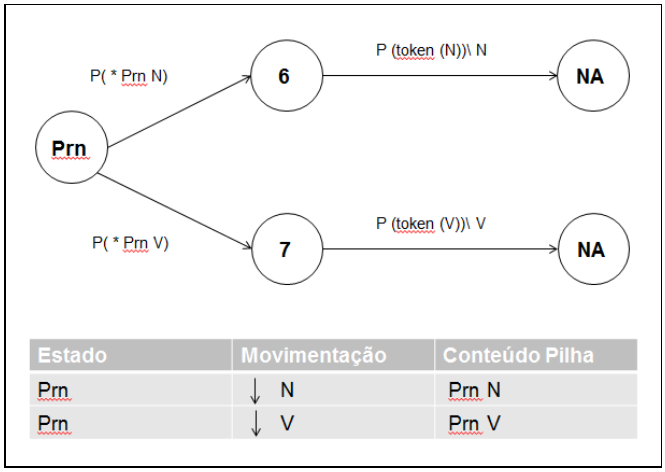
A Fig. 6 mostra como o autômato identifica e monta o primeiro sintagma. A coluna Movimentação indica que uma regra de redução foi encontrada na primeira pilha do vetor e o sintagma SN foi formado a partir dos dois elementos armazenados anteriormente (A N). O conteúdo da pilha é atualizado com a remoção dos símbolos A e N, e o símbolo SN é armazenado em seu lugar. Em seguida, SN é devolvido para a função adaptativa (δ (i)), que por sua vez cria o estado SN e uma nova função (η (i)) para continuar o reconhecimento a partir dele. Nota-se que a pilha A V não sofreu alteração e que continua ativa representando uma ramificação válida até este momento.



Estado	Movimentação	Conteúdo Pilha
N	A N ⟶ SN	SN
N	↑ SN	SN
V		A V

Figura 6. Formação do Sintagma SN

A Fig. 7 representa caminho seguido pelo autômato a partir do estado Prn. A movimentação é análoga àquela apresentada na Fig. 4. A função adaptativa (γ (i)) cria os estados 6, 7 e NA, este último usado para indicar que a sequencia analisada não é válida, o que ocorre quando a probabilidade do trigrama e da classificação do *token* combinados serem zero.



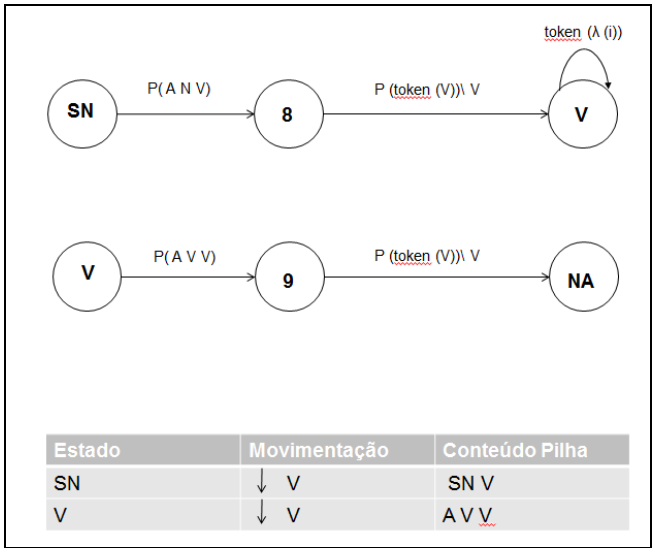
Estado	Movimentação	Conteúdo Pilha
Prn	↓ N	Prn N
Prn	↓ V	Prn V

Figura 7. Reconhecimento a partir de Prn

É o que ocorre com o trigrama * Prn N e o *token* "casa" classificado como substantivo. Embora "casa" possa ser um substantivo, de acordo com as regras gramaticais vigentes, nunca um substantivo vem antecedido de um pronome iniciando uma frase. No caso da segunda ramificação, chega-se a mesma conclusão com relação ao trigrama * Prn V e o *token* "casa" classificado como verbo.

A Fig.8 apresenta o reconhecimento a partir dos estados SN e V e do *token* "estava", classificado como verbo. A função adaptativa (η (i)) cria os estados 8 e V, indicando nas transições a probabilidade de ocorrência do trigrama A N V e do *token* "estava" ser classificado como verbo. O símbolo V é armazenado na pilha na qual se encontra SN e uma nova função (λ (i)) é criada para continuar o reconhecimento a partir de V.

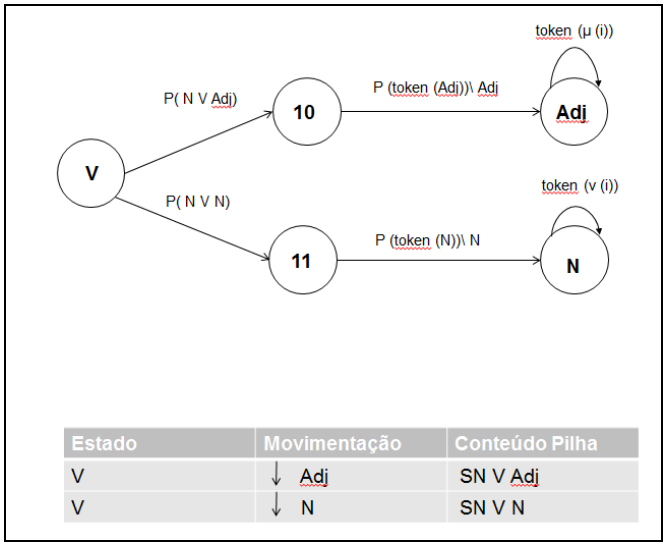
Por outro lado, partindo de V, a função (ζ (i)) cria os estados 9 e NA, indicando que o trigrama A V V não é valido. Ao final, o símbolo Adj é armazenado na fila correspondente e a função (μ (i)) é criada para continuar o reconhecimento a partir de Adj. Na outra ramificação, representa-se o reconhecimento do trigrama N V N e da probabilidade do *token* "estava" ser classificado como substantivo. Ao final, o símbolo N é armazenado na fila correspondente e a função (v (i)) é criada para continuar o reconhecimento a partir de N.



Estado	Movimentação	Conteúdo Pilha
SN	↓ V	SN V
V	↓ V	A V V

Figura 8. Reconhecimento a partir dos estados de SN e V

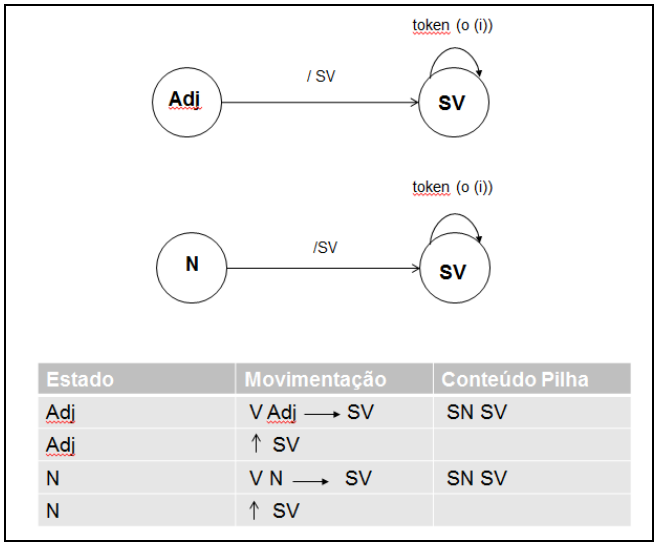
A Fig.9 apresenta o reconhecimento a partir do estado V e do *token* "aberta". Neste caso, o *token* pode ser classificado como adjetivo ou, menos usualmente, como substantivo. A função adaptativa (λ (i)) cria os estados 10, 11, Adj e N. Na ramificação V 10 Adj é representado o reconhecimento do trigrama N V Adj, assim como a probabilidade do *token* "aberta" ser classificado como adjetivo. Ao final, o símbolo Adj é armazenado na fila correspondente e a função (μ (i)) é criada para continuar o reconhecimento a partir de Adj. Na outra ramificação, representa-se o reconhecimento do trigrama N V N e da probabilidade do *token* "estava" ser classificado como substantivo. Ao final, o símbolo N é armazenado na fila correspondente e a função (v (i)) é criada para continuar o reconhecimento a partir de N.



Estado	Movimentação	Conteúdo Pilha
V	↓ Adj	SN V Adj
V	↓ N	SN V N

Figura 9. Reconhecimento a partir de V

A Fig. 10 apresenta o reconhecimento a partir dos estados Adj e N. A coluna Movimentação indica que uma regra de redução foi encontrada, gerando o sintagma SV a partir dos símbolos V e Adj, que são removidos da pilha. Em seguida, o símbolo SV é armazenado na pilha e a função (μ (i)) é notificada, criando o estão SV e a função adaptativa (o (i)). Na outra ramificação, o procedimento é análogo. O estado final SV é o mesmo nos dois casos e a função adaptativa responsável pelo reconhecimento a partir de SV também.

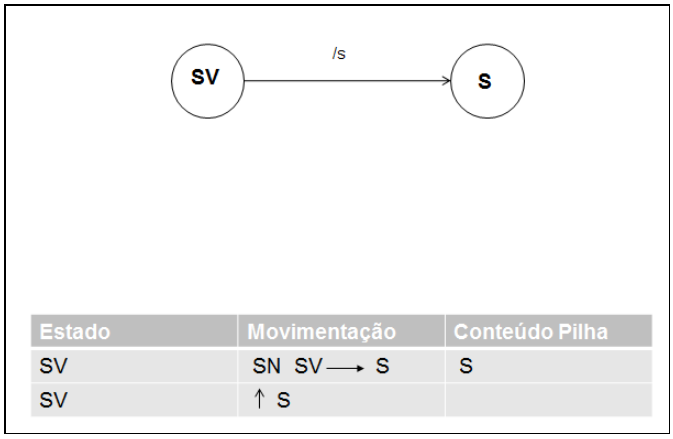


Estado	Movimentação	Conteúdo Pilha
Adj	V Adj ⟶ SV	SN SV
Adj	↑ SV	
N	V N ⟶ SV	SN SV
N	↑ SV	

Figura 10. Reconhecimento a partir de Adj e N

A Fig.11 apresenta a última etapa do reconhecimento. A coluna Movimentação indica que uma regra de redução foi encontrada, gerando o sintagma S a partir dos símbolos SN e SV, que são removidos da pilha. Em seguida, o símbolo S é armazenado na pilha e a função (o (i)) é notificada, criando o estado final S. As movimentações, assim como as probabilidades usadas para cálculo de cada transição podem ser recuperadas através de um transdutor vinculado ao autômato.

Ao final da movimentação, caso o autômato não chegue a um estado de aceitação, a frase é considerada incorreta pela gramática utilizada.



Estado	Movimentação	Conteúdo Pilha
SV	SN SV ⟶ S	S
SV	↑ S	

Figura 11. Etapa final

Considerações Finais

Este artigo apresentou conceitos de processamento da língua natural e uma revisão bibliográfica dos trabalhos realizados sobre o tema. Em seguida, foi proposto um modelo que unifica métodos determinísticos, estatísticos e heurísticos, utilizando autômatos adaptativos como tecnologia subjacente.

Referências

[1] JURAFSKY, D.; MARTIN, J. H.. Speech and Language Processing. 1024 p. Prentice Hall, 2000.

[2] PRICE, D. et al. Natural Java: A Natural Language Interface for Programming in Java. Proceedings of the 5th international conference on intelligent user interfaces. New Orleans, Louisiana, United States. Pages: 207 – 211, 2000. ISBN:1-58113-134-8.

[3] NUNES, M.G.V.; OLIVEIRA, O.N. O processo de desenvolvimento do Revisor Gramatical ReGra. Anais do XXVII SEMISH (XX Congresso Nacional da Sociedade Brasileira de Computação), 2000, Volume 1, p.6

[4] NAVIGLI, R.;VELARDI, P.; GANGEMI, A. Ontology Learning and its Application to Automated Terminology Translation. IEEE Intelligent Systems, Volume 18 Issue 1. Publisher: IEEE Educational Activities Department , 2003.

[5] GRISHMAN, RALPH. Information Extraction: Techniques and Challenges. Lecture Notes In Computer Science; Vol. 1299. International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, 1997.

[6] RILOFF, ELLEN; LORENZEN JEFFREY. Extraction-based text categorization: Generating Domain Specific role relationships automatically. In Natural Language Information Retrieval, Dordrecht, The Netherlands: Kluwer Academic Publishers, 1999. p. 167-196.

[7] LLORÉNS, JUAN; ASTUDILLO, HERNÁN. Automatic generation of hierarchical taxonomies from free text using linguistic algorithms. Lecture Notes in Computer Science Vol. 2426. Proceedings of the Workshops on Advances in Object-Oriented Information Systems, 2002.

[8] 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.

[9] RICH,E.; KNIGHT, K. Inteligência Artificial, 2. Ed. São Paulo: Makron Books, 1993.

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

[11] LADEIRA, M.P. Processamento da Linguagem Natural: Caracterização da Produção Científica dos Pesquisadores Brasileiros. Tese de Doutorado, UFMG, Minas Gerais, 2010.

[12] NETO, J. J.; MENEZES, C. E. D. Um Método para a Construção de Etiquetadores Morfológicos Aplicado a Língua Portuguesa, baseado em Autômatos Adaptativos. In: PROPOR 2000 – V Encontro para o Processamento Computacional da Língua Portuguesa, 2000, Atibaia. Anais do V Encontro para o Processamento Computacional da Língua Portuguesa. São Carlos : ICMS-USP, 2000. p. 53-64.

[13] PADILHA, E. G. ; VICCARI, R. M. Morfologia da Língua Portuguesa com Máquinas de Estados Finitos. In: 5o. Workshop de Processamento da Língua Portuguesa Falada e Escrita (PROPOR-2000), 2000, Atibaia. Anais do 5o. PROPOR - Workshop de Processamento da Língua Portuguesa Falada e Escrita, 2000.

[14] BONFANTE, A. G.; NUNES, M. G. V. Parsing Probabilístico para o Português do Brasil. In: I Workshop de Teses e Dissertações em Inteligência Artificial (I WTDIA), 2002, Porto de Galinhas - Recife. I Workshop de Teses e Dissertações em Inteligência Artificial (I WTDIA), 2002.

[15] BICK, E. The Parsing System PALAVRAS: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework, 2000. Ph. D. Thesis, Arhus University.

[16] JULIA, R. M. S.; SEABRA, J. R.; SEMEGHINI-SIQUEIRA, I. An Intelligent Parser that Automatically Generates Semantic Rules during Syntactic and Semantic Analysis. In: IEEE International Conference on Systems, Man and Cybernetics, 1995, Vancouver. v. i. p. 806-811.

[17] PIAGET, J. A construção do real na criança. Trad. Álvaro Cabral. Rio de Janeiro: Zahar, 1970. 360p.

[18] CHOMSKY, NOAM. Three models for the description of language. IRE Transactions on Information Theory 2: 113-124, 1956.

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

[20] NUNES et al. Introdução ao Processamento das Línguas Naturais. Notas didáticas do ICMC Nº 38, São Carlos, 88p, 1999. Paris, Hachette, 1992. 158p.

[21] KARLSON, F. Constraint Grammar as a Framework for Parsing Running Text. 13o. International Conference on Computational Linguistics, 1990, Helsinki (Vol.3, p.168-173).

[22] MARQUES, N.M.C.; LOPES, J.G.P. Redes Neuronais e um Léxico na Etiquetagem Morfossintática para o Estudo da Subcategorização Verbal. In: SARDINHA, T. B. A Língua Portuguesa no Computador. 295p. Mercado de Letras, 2005. P. 71-90.

[23] DIAS, G.H.; LOPES, J.G.P. Extração Automática de Unidades Poli lexicais para o Português. In: SARDINHA, T. B. A Língua Portuguesa no Computador. 295p. Mercado de Letras, 2005. P. 155-184.

[24] NETO, J.J. Laboratório de Linguagens e Tecnologias Adaptativas, 2004. Disponível em: < http://lta.poli.usp.br/lta/roteiro-de-estudos >. Acesso: 05/06/2014

[25] CONTIER, A.; PADOVANI, D.; NETO,J.J.. Linguístico: Usando Tecnologia Adaptativa para a Construção de Um Reconhecedor Gramatical. Em: Memórias do VIII Workshop de Tecnologia Adaptativa – WTA 2014. EPUSP, São Paulo, ISBN: 978-85-86686-76-4, pp. 8-20. 06 e 07 de Fevereiro de 2014.

[26] http://www.linguateca.pt/

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

[28] LUFT, C.. Moderna Gramática Brasileira. 2ª. Edição Revista e Atualizada. 265p. Editora Globo, 2002.

Djalma Padovani nasceu em São Paulo em 1964. Cursou bacharelado em Física pelo Instituto de Física da Universidade de São Paulo, 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 é responsável pela arquitetura tecnológica da Serasa S/A, empresa do grupo 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.