Armazenador Adaptativo de Palavras e Frases da Língua Portuguesa

(19 Dezembro 2008)

M. Marques

***Resumo*— Neste trabalho apresentaremos uma proposta de aplicativo adaptativo para tratamento e armazenamento de palavras ou frases da língua portuguesa, demonstrado através de teste a um conjunto limitado de frases. O aplicativo proposto é baseado nos conceitos de ambiente colaborativo de aprendizado por descoberta (Collaborative Discovery Learning Environment - CDLE) e no framework de dispositivos adaptativos dirigidos por regras (DADR). O CDLE permite a construção de um conhecimento próprio, por meio da execução de experimentos dentro de um domínio e por meio da inferência e adição de novas regras ou informações gramaticais. Também é proposto um novo paradigma de programação nomeado como programação orientada a conceito (POC).**

***Palavras chave*— Ambiente de Descobrimento e Aprendizado Colaborativo, Dispositivos Adaptativos Dirigidos por Regras, Processamento de Linguagem Natural, Tabela de Decisão Adaptativa.**

I. Introdução

Este documento descreve um aplicativo adaptativo para processamento de linguagem natural, mais especificamente para o português do Brasil. A base do aplicativo é o conceito de CDLE, como apresentado em [4], por meio do qual o aplicativo aprende novas construções gramaticais ou novas palavras e as incorpora em seu ambiente de conhecimento, por meio da produção de novas regras de reconhecimento de padrões. Para permitir o processo de aprendizagem, o aplicativo efetua a comparação das frases que ele recebe em seu ambiente de interação com o usuário, com as regras previamente definidas no aplicativo adaptativo. As regras são baseadas no *framework* DADR, que foi introduzido primeiramente em [4] e compreende um formalismo não adaptativo básico e um formalismo adaptativo que altera o funcionamento do autômato adaptativo, conforme definido por Neto como M = (ND, AM), em que ND é o mecanismo não adaptativo dirigido por regras pré-definidas e AM é o mecanismo adaptativo. Caso as regras pré-definidas ou incorporadas dinamicamente no aplicativo não consigam tratar a nova frase recebida como uma frase padrão do conhecimento já acumulado, o aplicativo fará a opção por inferir uma nova regra.

O restante do artigo é organizado como segue. A seção 2 descreve as principais características do aplicativo adaptativo para processamento de linguagem natural. A seção 3 apresenta a tabela de decisão adaptativa que é gerada para reconhecimento das frases utilizadas nos testes da seção 2. A seção 4 apresenta o diagrama padrão de funcionamento da aplicação adaptativa para processamento de linguagem natural. A seção 5 apresenta a conclusão do artigo e perspectivas de trabalhos futuros que podem ser feitos.

M. Marques exerce a docência no Departamento de Computação da Faculdade Sumaré e na Escola Técnica de São Paulo, que pertence ao Centro Paula Souza e pode ser contactado no email mario.marques@sumare.edu.br.

II. Características principais do aplicativo adaptativo

O aplicativo que foi desenvolvido tem a finalidade de validar o conceito de que é possível criar um mecanismo de reconhecimento de frases e dessa forma poderia ser classificado na categoria de armazenador de palavras, que precede ao desenvolvimento do módulo de reconhecimento gramatical, conforme definido em [3]. Os reconhecedores gramaticais treináveis, conforme em [2] são compostos de três módulos, mas precedidos de uma fase inicial de treinamento do autômato, em que ele é exposto a um corpus anotado para treinamento. Após a etapa de treinamento, o primeiro módulo efetua a atividade de etiquetar as palavras conhecidas, a partir do aprendizado realizado no corpus anotado. O segundo módulo efetua a marcação das palavras desconhecidas e o terceiro módulo faz o refinamento contextual. A diferença que existe entre o reconhecedor gramatical proposto por Brill, que efetua a classificação por meio de etiquetas de cada frase reconhecida e a proposta aqui apresentada, é a substituição do módulo de treinamento em corpus anotado, por um módulo de armazenamento de palavras, que serão obtidas por meio de interação com o usuário do ambiente CDLE. No caso em que a palavra já seja conhecida não será necessário acionar o módulo de marcação morfológica, mas se a palavra for desconhecida será acionado o primeiro módulo, que contém regras pré-definidas para efetuar a marcação morfológica da palavra. Caso não haja uma referência direta entre a nova palavra e as regras existentes no módulo 1, a palavra será encaminhada para o módulo 2 que por meio de consulta ao usuário efetuará a marcação da palavra e a inserção da palavra na lista de palavras conhecidas. Nos casos em que a palavra passe pelo módulo 2, a mesma ainda será submetida ao módulo 3 para verificação contextual.

O aplicativo foi construído apenas com uma estrutura sintática, a qual permite o reconhecimento de uma única estrutura de frase simples do português do Brasil e que corresponde a porção de regras fixas ou ND do autômato.. A frase que será reconhecida deve ter apenas três componentes: um sujeito como elemento inicial, um verbo como segundo elemento e um complemento como terceiro elemento, conforme definido por:

Frase → {sujeito verbo complemento}

A gramática implementada para reconhecimento de frases, prevê sempre a ocorrência de apenas um único elemento da gramática: sujeito (s), verbo (v) e complemento (c) e está definida por:

$L = \{\mathbf{F} \in \{s,v,c\} \mid \mathbf{F} = s,v,c,\ n=1\}$

O autômato que foi especificado para reconhecimento da gramática, prevê a existência de quatro símbolos possíveis:

- S que representa a transição para estado vazio;
- s que representa o elemento sujeito da frase;
- v que representa o verbo da frase;
- c que representa o complemento da frase.

Uma frase para ser considerada completa deve possuir os três elementos (s,v,c), separados por transições vazias da seguinte forma:

$G = (\{S, s,v,c\}, \{s,v,c,\}, \{S \rightarrow \epsilon, S \rightarrow sSvSc\}, S)$

Para realizar os testes com o aplicativo adaptativo foram utilizadas as seguintes frases:

Pássaro é azul.

Pássaro é verde.

O pássaro é bonito.

A partir da inserção da frase "Pássaro é azul", o autômato constrói a estrutura de reconhecimento especificada na figura 1.

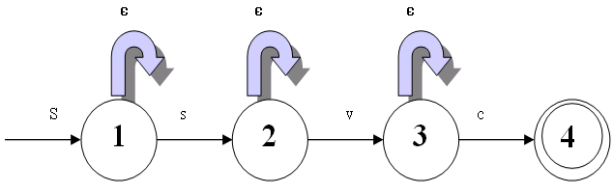

Fig. 1. O autômato em ação.

A partir da inserção da frase "Pássaro é verde" o autômato utiliza-se da mesma estrutura de reconhecimento especificada na figura 1.

A partir da próxima frase que é "O pássaro é bonito" o autômato terá dificuldades para reconhecer esta frase, pois ela contém quatro elementos e por este motivo solicitará o auxílio do usuário. A idéia básica é tentar reconhecer se as palavras da frase correspondem a palavras pré-cadastradas em frases anteriores e já catalogadas em categorias gramaticais componentes da estrutura da frase padrão. Dessa forma, ao analisar a cadeia de entrada, o autômato, por comparação, verifica que ele já conhece a palavra pássaro, que ele já catalogou como um substantivo e sujeito da frase, a palavra é que foi catalogada como um verbo, e o autômato não reconhece as palavras o e bonito. Com as informações que já são de conhecimento do autômato ele pergunta ao usuário qual é a categoria gramatical da palavra "o".

Como a resposta será a categoria de adjunto adnominal e artigo, o autômato então gera uma nova regra de reconhecimento, que corresponde à porção AM do autômato, em que é inserido o elemento a para representar a existência de um novo elemento, anterior ao substantivo, e que pode ser preenchido ou pode também aparecer vazio, conforme definido por:

$L = \{F \in \{a,s,v,c\} \mid F = s,v,c,\ a=*\ e\ s,v,c=1\}$

Uma nova gramática também é gerada, com a inclusão da possibilidade da existência de um elemento anterior. Dessa forma o autômato pode reconhecer frases que sejam formadas por três (s,v,c) ou quatro elementos (a,s,v,c), separados por transições vazias, conforme:

$G = (\{S, a, s, v, c\}, \{a, s, v, c,\}, \{S \rightarrow \epsilon, S \rightarrow aSsSvSc\}, S)$

A nova máquina de reconhecimento é ilustrada na figura 2, em que o elemento a é opcional na verificação da frase.

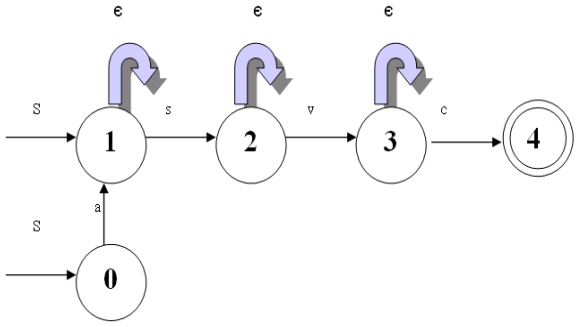

Fig. 2. O novo autômato.

Após a realização da ação adaptativa as novas palavras aprendidas devem ser armazenadas no léxico construído para posterior utilização. Specia e Rino [6] relatam à existência de duas abordagens mais usualmente utilizadas no armazenamento de itens lexicais, do ponto de vista da tecnologia da informação: base de dados ou arquivos seqüenciais. Do ponto de vista lingüístico as opções apontadas em [6] correspondem às citadas formas canônicas e as formas originais. No modelo em que a forma canônica é armazenada é definido um processo em que a partir da forma canônica são geradas as formas analisadas, com as devidas flexões de gênero, número, espécie, grau, modo, tempo, etc. Já no modelo de armazenamento das formas originais todas as variações de cada palavra são armazenadas. Neste trabalho, do ponto de vista lingüístico, utilizamos o armazenamento das formas canônicas das palavras e do ponto de vista computacional propomos um novo método, baseado no conceito de programação orientada a conceito (POC).

A POC é um novo paradigma para desenvolvimento de programas computacionais, pois propõe a identificação de cada palavra da língua portuguesa como um elemento componente da linguagem de programação POC. Em seu modelo conceitual, a POC é bastante similar à especificação de programação orientada a objetos (POO). As estruturas básicas da POC são

- Estrutura;
- Conceito.

A estrutura é a forma de implementação das estruturas de frase aprendidas pelo autômato adaptativo e, para cada nova estrutura de frase aprendida, uma nova estrutura é criada automaticamente pelo aplicativo adaptativo e carregada, automaticamente, como parte integrante do conjunto de frases conhecidas.

O conceito corresponde à especificação dos diferentes conceitos que podem ser aprendidos pelo aplicativo adaptativo. Cada conceito é composto de 4 elementos:

- Nome do conceito;
- Métodos;
- Atributos;
- Definição.

Cada conceito deve possuir um nome, por exemplo, o conceito de homem terá o nome homem. Além disso, cada conceito em POC possui métodos e atributos, que guardam estreita similaridade com os seus correspondentes em POO. Já a definição do conceito identifica de forma única ou coletiva cada conceito e também cada conceito deve pertencer a um conjunto. Por definição, a POC possui dois grandes conjuntos : o conjunto dos seres conscientes (C) e o conjunto dos seres não-conscientes (NC).

A categorização dos elementos da frase (conceitos) com o uso de teoria dos conjuntos, tem como base a definição de uma função, que permita localizar a qual conjunto a palavra da frase pertence. Por exemplo, a frase "O pássaro tem penas", deve ser definida por uma função f:A→B, biunívoca e definida sobre B. Os elementos da frase são inseridos no conjunto A, ou seja, A= {o, pássaro, tem, penas} e o conjunto B como B= {C, NC}, de modo que a função seja f(x) =x, sendo que se o x resultante for maior do que 1, x>1, ele corresponde ao conjunto NC, subconjunto não-animado (NA), caso x<=1, temos duas possibilidades, sendo que podemos classificá-lo como C ou NC e animado (A). Dessa forma, todas as palavras aprendidas pelo autômato serão classificadas e armazenadas, permitindo o tratamento de ambigüidade lingüística.

III. A TABELA DE DECISÃO ADAPTATIVA

A tabela de decisão adaptativa (TDA), conforme definida em [5] é o mecanismo utilizado para implementar o controle das ações que serão executadas por um dispositivo computacional que é composto por duas partes: um conjunto de regras pré-definidas e um conjunto de ações adaptativas, as quais podem ser representadas pela fórmula TDA = (ND, AM). Como já descrito anteriormente, a porção ND correspondem às regras não-adaptativas e a porção AM correspondem às ações adaptativas.

Na figura 3 temos o modelo conceitual da tabela de decisão adaptativa, conforme definido por Neto. Pode-se verificar que a tabela é composta das regras pré-definidas na parte superior da tabela relacionadas como tabela de decisão subjacente e as ações adaptativas na parte inferior da tabela relacionadas como funções adaptativas.

Cada coluna deve receber uma letra que indica qual tipo de ação deverá ser executada naquela coluna e na figura 3, extraída de [8], esta informação está identificada com o nome Tag. Os tipos de Tag possíveis são H ? + - S R E.

O Tag H indica o cabeçalho da função adaptativa e pode conter os seguintes caracteres:

A, para indicar se a regra deve ser executada antes da função adaptativa subjacente;

B, para indicar se a regra deve ser executada depois da função adaptativa subjacente;

P, para indicar um parâmetro formal;

V, para indicar uma variável;

G, para indicar um gerado.

Os caracteres ? + - indicam respectivamente as funções de consulta, adição e exclusão de uma ação adaptativa.

As colunas com os caracteres S R E indica respectivamente a primeira ação que deverá ser executada, a execução de uma regra normal não-adaptativa e o final da execução do dispositivo.

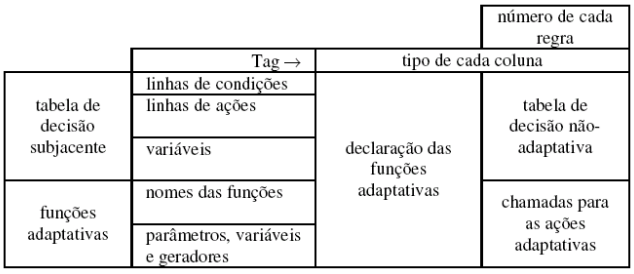

Fig. 3. Modelo conceitual da tabela de decisão adaptativa.

IV. DESCRIÇÃO DO APLICATIVO

O aplicativo que foi desenvolvido para validar a aplicabilidade do modelo aqui proposto foi construído na linguagem de programação Java, portanto baseado no paradigma de programação orientada a objeto, mas com algumas adaptações, devido à inexistência de uma linguagem de programação POC.

O aplicativo é composto de uma interface gráfica, que permite a interação do usuário com o módulo de interpretação de linguagem natural, para viabilizar, ao usuário, a digitação de frases e a resposta do usuário a questões colocadas pelo aplicativo.

O aplicativo desenvolvido neste artigo é o primeiro módulo da proposta de construção de um aplicativo de aprendizado e interação em linguagem natural, que deverá ser composto dos seguintes módulos: (conforme ilustrados na figura 4)

- Interface com o usuário - é o ambiente computacional em que o usuário pode interagir com o armazenador adaptativo, por meio da digitação de frases.
- Armazenador - cada frase digitada é avaliada para verificação da estrutura da frase e reconhecimento das palavras digitadas.
- Refinador Contextual - verifica a melhor maneira de classificar a estrutura da frase e das palavras, caso seja necessária à avaliação de novas estruturas ou palavras desconhecidas.
- Analisador Morfológico - contem as regras iniciais de classificação morfológica das palavras e, por meio de interação com o refinador contextual, pode ter novas regras inseridas através de processo de aprendizado adaptativo.
- Analisador Sintático e Analisador Semântico - tem o mesmo comportamento e interação do módulo Analisador Morfológico.
- Gerador de Produções - responsável por avaliar e produzir as frases que serão enviadas ao usuário, com o intuito de obter mais informações sobre frases recebidas do usuário ou para expressar as informações armazenadas.

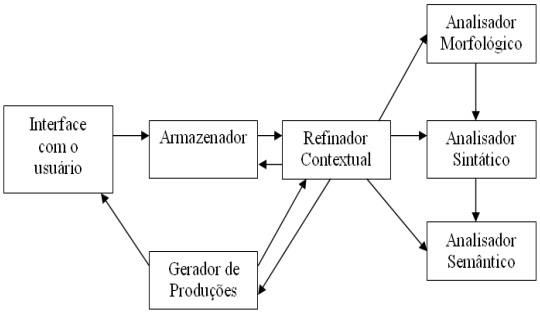

Fig. 4. Módulos do aplicativo adaptativo de aprendizado de linguagem natural.

O aplicativo é composto pela classe PhraseAnalyzer que efetuará primeiramente a separação das palavras da frase digitada por meio de seu método slicer. Após obter a quantidade de palavras da frase o método queryStructure fará a verificação da existência da estrutura recebida no aplicativo adaptativo. Caso a quantidade de elementos da frase seja incompatível com a estrutura pré-armazenada inicialmente no dispositivo (3 elementos), será acionado, automaticamente, o método createNewStructure. Este método executará a ação adaptativa exemplificada na figura 2 e criará uma nova estrutura que permitirá o armazenamento de uma frase com mais elementos (no caso do exemplo utilizado neste trabalho 4 elementos). Nos casos em que a estrutura da frase já exista ou no caso em que seja necessário criar um nova estrutura, as palavras da frase serão encaminhadas ao método queryWord para verificação da existência das palavras no aplicativo adaptativo e caso alguma palavra ainda não seja conhecida pelo aplicativo será acionado o método createNewWord que fará a inclusão da nova palavra no aplicativo.

O algoritmo do aplicativo, com o uso da tabela de decisão adaptativa, é o seguinte:

1) estando na configuração inicial da tabela executa a regra S;

2) enquanto houver símbolo a ser lido na cadeia de entrada faça;

2.1) executa a divisão da frase em seus componentes;

2.2) procura regra aplicável;

2.2.1) Se existir regra aplicável aceita a cadeia, senão executa o passo 2.5;

2.3) verifica se as palavras são todas conhecidas.

2.3.1) Se existir palavra desconhecida inclui a palavra na lista de palavras conhecidas;

2.4) fim enquanto.

2.5) executa a função adaptativa;

2.5.1) inclui nova regra;

2.5.2) volta ao passo 2.2;

V. V. CONCLUSÃO E PERSPECTIVAS FUTURAS

A partir dos experimentos realizados observou-se que os resultados obtidos foram adequados ao esperado. Os testes realizados contaram com frases de apenas 3 ou 4 elementos, e nos dois casos o aplicativo conseguiu verificar as palavras de cada frase e efetuar o armazenamento de novas palavras e gerar uma nova estrutura de reconhecimento de frase com 4 elementos.

A programação foi efetuada com a utilização da linguagem de programação Java devido a sua ampla utilização, por ser código portável e permitir a reutilização em trabalhos futuros. Trabalhos ainda precisam ser feitos para especificar e desenvolver uma linguagem que implemente a POC.

Cabe ressaltar que foi desenvolvido no presente trabalho apenas os módulos referentes à interface com o usuário e o módulo armazenador, ficando para trabalhos futuros o desenvolvimento dos módulos refinador contextual, morfológicos, sintáticos e semânticos.

As perspectivas de trabalhos futuros e potencial da solução aqui modelada e desenvolvida são muito grandes, podendo ampliar o escopo de outras soluções similares já desenvolvidas e que propõe sempre a identificação de palavras por meio de comparação com corpus anotado, sendo que a presente proposta sugere a possibilidade de iniciar o processo de identificação de palavras a partir do aprendizado de novas regras e novas palavras, por meio da interação direta com o usuário, tornando o processo de aprendizado mais próximo ao análogo processo de aprendizado humano.

REFERÊNCIAS

Periodicals (Artículos de revista):

[1] B.F.T. Azevedo e O.L. Tavares Um ambiente inteligente para aprendizagem colaborativa. XII SBIE 2001 – Simpósio Brasileiro de Informática na Educação, UFES, 2001.

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

[3] C.E.D. Menezes e J.J. Neto. Um método híbrido para a construção de etiquetadores morfológicos, aplicado à língua portuguesa, baseado em autômatos adaptativos. Anais da Conferencia Iberoamericana en Sistemas, Cibernética e Informática, 19-21 de Julio, 2002, Orlando, Florida.

[4] J.J..Neto Uma solução adaptativa para reconhecedores sintáticos. Technical report, PCSPOLI-USP, São Paulo, Brasil, 1988.

[5] J.J. Neto. Adaptive Rule-Driven Devices - General Formulation and Case Study. Lecture Notes in Computer Science. Watson, B.W. and Wood, D. (Eds.): Implementation and Application of Automata 6th International Conference, CIAA 2001, Vol.2494, Pretoria, South Africa, July 23-25, Springer-Verlag, 2001, pp. 234-250.

[6] L. Specia e L.H.M. Rino, O desenvolvimento de um léxico para a geração de estruturas conceituais UNL. Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional – NILC-TR-02-14. São Carlos, SP, setembro de 2002.

[7] C. Y. O. Taniwaki, e J.J. Neto. Autômatos Adaptativos no Tratamento Sintático de Linguagem Natural Boletim Técnico PBT/PCS, Escola Politécnica, São Paulo, 2001

[8] A. Tchemra, Tabela de decisão adaptativa:simulação de um autômato adaptativo. WTA 2008 – Segundo Workshop de Tecnologia Adaptativa. São Paulo, 2008, págs. 5-8.

Mario Marques nasceu em São Paulo, em 7 de Outubro de 1967. Graduou-se na Universidade Mackenzie e na Universidade de São Paulo e formou-se mestre no IPT.

Exerce atividades profissionais na Caixa Economica Federal e atividades de ensino na Escola Técnica Estadual de São Paulo e na faculdade Sumaré.

Entre seus campos de interesse estão o aprendizado automático de conhecimento, processamento de linguagem natural e máquinas adaptativas.