# Aplicação de Aprendizado Baseado em Transformações na Identificação de Sintagmas Nominais

**Cícero Nogueira dos Santos**[1], **Claudia Oliveira**[2]

[1] Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rio de Janeiro, RJ – Brasil

nogueira@inf.puc-rio.br

[2]Departamento de Sistemas e Computação
Instituto Militar de Engenharia (IME)
Rio de Janeiro, RJ – Brasil

cmaria@centroin.com.br

***Abstract.*** *This paper describes a case study of Portuguese NP identification with the TBL framework. Within the TBL framework, we propose a new type of rule template that makes possible the generation of more efficient rules for the classification of prepositions. In the identification of Brazilian Portuguese NPs with the developed TBL tool, the best result obtained was 86,6% for precision and 85,9% for recall.*

***Resumo.*** *Este artigo descreve um estudo de caso da aplicação do método TBL para a identificação de Sintagmas Nominais de textos em português. Dentro do contexto do método TBL foi proposto um novo tipo de molde de regras que possibilitou a geração de regras mais eficazes para a classificação específica das preposições. Na identificação de SNs do português brasileiro com o uso da ferramenta TBL desenvolvida foi obtido, no melhor caso, precisão de 86,6% e abrangência de 85,9%.*

## 1. Introdução

Nos últimos anos, tem-se observado um retorno aos métodos empíricos de Processamento de Linguagem Natural, em que a aquisição do conhecimento pela máquina é majoritariamente realizada com base nos dados. Desde a última década, várias técnicas de Aprendizado de Máquina (AM) têm sido utilizadas para a identificação automática de sintagmas nominais (SNs). A identificação de SNs em textos tem aplicações em diversos problemas como: recuperação e extração de informações, análise sintática, resolução de co-referência, identificação de relações semânticas, entre outros. A maior parte dos trabalhos publicados na área têm o inglês como língua alvo; trabalhos sobre o uso de técnicas de AM para a identificação de SNs do português brasileiro não foram encontrados.

Neste trabalho, a identificação automática de SNs do português brasileiro é tratada como uma tarefa de classificação a ser automaticamente aprendida com o uso da técnica de AM chamada Aprendizado Baseado em Transformações (do inglês *Transformation Based Learning* – TBL). Nesta abordagem, o aprendizado é guiado por um corpus de treino que contém exemplos corretamente classificados. A própria classificação dos

exemplos foi derivada automaticamente a partir de um corpus preexistente. O conhecimento lingüístico gerado por essa técnica consiste de uma lista ordenada de regras de transformação, que pode ser utilizada para a classificação de novos textos.

O restante deste trabalho está organizado da seguinte forma: na seção 2. são mostrados alguns conceitos sobre SNs e a identificação automática de SNs; na seção 3. é mostrado o funcionamento do algoritmo TBL; na seção 4. é apresentado alguns aspectos da ferramenta TBL desenvolvida; na seção 5. são mostrados os experimentos e resultados.

## 2. Identificação de SNs

O *sintagma* consiste num conjunto de elementos que constituem uma unidade significativa dentro da sentença e que mantêm entre si relações de dependência e de ordem. Organizam-se em torno de um elemento fundamental, denominado núcleo, que pode, por si só, constituir o sintagma. Segundo [Lobato 1986] os sintagmas são formados por constituintes, que são um conjunto de palavras de uma sentença que a divide, normalmente, em duas subpartes básicas: sintagma nominal (SN) e sintagma verbal (SV). Quando o núcleo for um verbo, tem-se um SV e quando o núcleo for um nome têm-se um SN. Funcionando como modificador de um SN ou de um SV, temos o sintagma preposicionado (SP), que combina preposições e substantivos.

Segundo [Perini 2003], o SN pode ser definido de maneira muito simples: "é o sintagma que pode ser sujeito de alguma oração". Por exemplo, na seguinte oração:

[Esse professor] é [um neurótico]

temos que "esse professor" é um SN porque é sujeito da oração; e "um neurótico" é também um SN, porque, mesmo que não seja sujeito nessa oração, pode ser sujeito em outra oração, como a seguinte:

[Um neurótico] rabiscou [meus livros]

A identificação automática de SNs diz respeito ao uso de programas computacionais que, dada uma sentença, possam identificar quais seqüências de palavras constituem SNs. Essa tarefa tem várias aplicações no campo de PLN, bem como em outras áreas como Recuperação de Informações (RI) e Extração de Informações (EI). Em PLN alguns dos usos seriam, por exemplo, para a resolução de co-referência; para a identificação de relações semânticas como hiperonímia/hiponímia; etc. A principal aplicação da identificação (e extração) de SNs em RI é a criação de termos de indexação.

### 2.1. SNs do português vs. SNs básicos do inglês

As línguas portuguesa e inglesa não são totalmente dissimilares no nível sintático. No geral, a ordem das palavras, com relação às classes gramaticais, é similar. Uma das principais diferenças em relação à estrutura do SN é que a posição do adjetivo é preferencialmente pré-nominal no inglês e pós-nominal no português.

Outra característica distinta do SN do inglês é que um substantivo pode ocupar a posição de um predicado adjetivo, como em "*door mat*", mas em português o mesmo tipo de SN precisa de uma preposição, como em "*tapete da porta*". A sequência *PREP (Det) N* é considerada como uma locução adjetiva por alguns gramáticos.

Na utilização de aprendizado de máquina para a identificação de SNs do inglês tem-se usado a noção de *SN básico,* que é definido por [Ramshaw and Marcus 1995]

como um SN não recursivo que inclui determinantes e pré-modificadores, mas não inclui pós-modificadores como sintagmas preposicionados e orações subordinadas. Em Português essa noção provê um conjunto de SNs muito pobre. Vejamos o seguinte exemplo de SN básico em inglês e a sua tradução para o português.

$_{SN}$[the first Government drug manufacturing plant]$_{SN}$
$_{SN}$[a primeira fábrica de $_{SN}$[ produção de $_{SN}$[ remédios do $_{SN}$[ governo] ] ] ]

Observa-se que a tradução tem necessariamente quatro SNs básicos aninhados, sendo que essa é uma construção bastante comum em português. Dessa forma, para se obter um SN mais "informativo" na lingua portuguesa, é necessário considerar SNs recursivos. Por isso, resolvemos identificar SNs que, em oposição aos SNs básicos, contenham pós-modificadores como adjetivos e sintagmas preposicionados, mas não incluindo pós-modificadores que contenham orações subordinadas. Logo, nesse modelo, o exemplo anterior em português representaria um único SN, como mostrado a seguir:

$_{SN}$[a primeira fábrica de produção de remédios do governo]$_{SN}$

Como resultado, o problema de identificar SNs do português torna-se mais complexo do que o da identificação de SNs básicos, visto que inclui o problema de ligação do sintagma preposicionado.

### 2.2. Codificação utilizada para identificar os SNs nas sentenças

A identificação de SNs é tratada como um problema de classificação, onde o objetivo é associar a cada item do corpus uma etiqueta adicional que o classifique como pertencente ou não a um SN. Nesse trabalho, é usado o conjunto de etiquetas {I,O,B} proposto por [Ramshaw and Marcus 1995], onde as palavras etiquetadas com I (*In*) pertencem a um SN, as marcadas com O (*Out*) estão fora de um SN, e a etiqueta B (*Begin*) é utilizada para marcar a palavra mais à esquerda de um SN que se inicia logo após um outro SN. Chamaremos essas etiquetas de *etiquetas SN*.

Nesse trabalho considera-se que o texto que terá seus SNs identificados já esteja etiquetado morfossintaticamente, ou seja, cada palavra já deve possuir uma etiqueta que identifica a sua classe de palavras.

A seguinte sentença, que tem os SNs identificados entre colchetes:

[O terrorismo] espalhou [medo] em [o mundo inteiro].

seria codificada como:

O/ART/I terrorismo/N/I espalhou/V/O medo/N/I em/PREP/O o/ART/I mundo/N/I inteiro/ADJ/O ./O

## 3. Aprendizado Baseado em Transformações

Nessa seção é descrito o funcionamento do algoritmo TBL. Para o entendimento desse método de aprendizado de máquina é fundamental a compreensão do conceito de *feature*, aqui traduzido como *traço*. Durante todo o trabalho, o termo *traço* denotará uma unidade observável de um item do corpus. Um item do corpus pode ser constituído por um ou mais traços e o tipo de tarefa é que define como esse item é representado e quais traços ele possui. Por exemplo, no caso da etiquetagem morfossintática (a tarefa de associar a cada palavra uma etiqueta que corresponde à sua classe de palavras) cada item é composto por dois traços: a unidade léxica (traço $word$) e a sua etiqueta morfossintática (traço $tpos$), como exemplificado a seguir:

O/ART rato/N comeu/V o/ART queijo/N.

Na tarefa de identificação de SNs, os itens são compostos por três traços: a unidade léxica, a sua etiqueta morfossintática e a etiqueta SN (traço $tsn$) que identifica se a palavra pertence ou não a um sintagma nominal:

O/ART/I rato/N/I comeu/V/O o/ART/I queijo/N/I.

### 3.1. O Algoritmo TBL

Aprendizado Baseado em Transformações (TBL) é um dos algoritmos de aprendizado de máquina baseados em regras mais bem sucedidos. Foi introduzido por Eric Brill [Brill 1995], e tem sido utilizado para diversas tarefas importantes de PLN tais como: etiquetagem morfossintática [Brill 1995]; identificação de sintagmas nominais do inglês [Ramshaw and Marcus 1995]; desambiguação da ligação do sintagma preposicionado; etiquetagem de atos de fala; análise sintática parcial [Ramshaw and Marcus 1995]; etc.

A idéia central do algoritmo TBL é gerar uma lista ordenada de regras que melhoram progressivamente uma classificação inicial atribuída aos itens do corpus de treino. O TBL é considerado como um algoritmo guloso, visto que, a cada iteração, a regra escolhida para ingressar na lista de regras aprendidas é aquela que provocar maior redução de erros na classificação atual dos itens do corpus de treino.

Para a utilização do algoritmo TBL são necessários os seguintes itens de entrada:

- um corpus de treino contendo a classificação correta para algum traço lingüístico que deseja-se aprender a classificar;
- uma classificador básico (*Base-Line System*), utilizado para atribuir uma classificação inicial aos itens do corpus, geralmente baseada em freqüências verificadas no corpus de treino;
- um conjunto de *moldes de regras (templates)*. Esses *moldes* determinam os tipos de expressões condicionais das regras geradas, indicando combinações de traços, na vizinhança de um item, que possam determinar a classificação desse item.
- uma função objetivo $f$ para o aprendizado. O tipo de função objetivo mais utilizada em TBL é a seguinte, que representa o ganho de performance resultante da aplicação de uma regra $r$:

  $$f(r) = good(r) - bad(r)$$

  onde:

  $good(r)$= Total de erros que $r$ corrige
  $bad(r) =$ Total de erros que $r$ provoca

  O valor de $f(r)$ será denotado por *pontuação* de $r$.

A Figura 1 mostra o processo de aprendizagem utilizado pelo método TBL. O aprendizado inicia-se com a atribuição de uma classificação inicial aos itens do corpus de treino com a utilização do classificador inicial (*base-line system*). Em seguida, a classificação resultante é comparada com a classificação correta e em cada ponto em que houver erro, todas as regras que o corrigem serão geradas a partir da instanciação dos moldes de regras com o contexto do item atualmente analisado. Normalmente uma regra $r$ irá corrigir alguns erros ($good(r)$), mas também poderá provocar outros erros pela alteração de itens que estavam classificados corretamente ($bad(r)$). Dessa forma, após computados os valores de $good$ e $bad$ para todas as regras candidatas, a regra que tiver

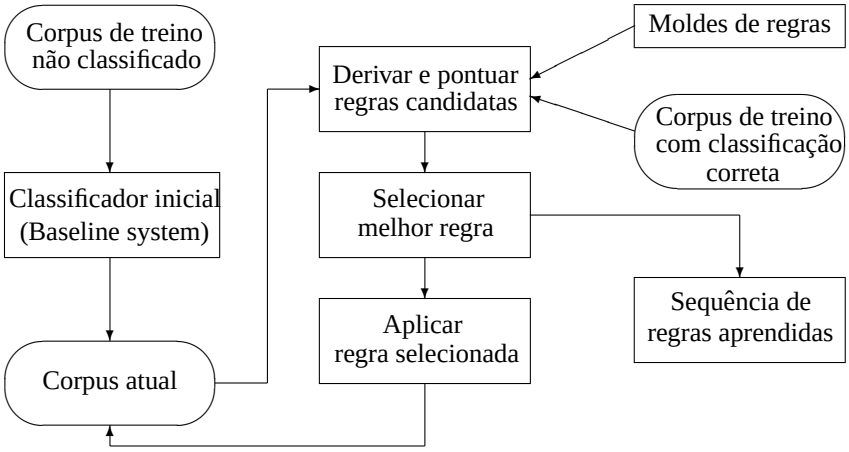


**Figura 1. Aprendizado Baseado em Transformações**

maior pontuação será selecionada e colocada na lista de regras aprendidas. A regra selecionada é então aplicada ao corpus, e o processo de geração de regras será reiniciado enquanto for possível gerar regras com pontuação acima de um limite especificado.

Na classificação de novos textos com uso dessa técnica necessitamos apenas submeter o texto ao classificador inicial, e logo em seguida aplicar a lista de regras na seqüência em que foram aprendidas.

### 3.2. Regras e Moldes de Regras

As regras contextuais geradas pelo método TBL seguem o formato:

$$\langle t_1 \rangle = val_1 \ \langle t_2 \rangle = val_2 \ \langle t_3 \rangle = val_3 \ \ldots \ \langle t_n \rangle = val_n \ \rightarrow \ \langle ftr \rangle = val$$

No lado esquerdo da seta existe uma expressão condicional formada pela conjunção de pares $\langle t_i \rangle = val_i$, onde $t_i$ é um *Termo Atômico* (TA) e $val_i$ é um valor válido para ele. Do lado direito da seta é indicada a associação do valor $val$ ao traço $ftr$. Uma regra aplica-se a um determinado item do corpus (item alvo), se nesse item $ftr \neq val$ e a expressão condicional for verdadeira. A expressão condicional é verificada substituindo-se os termos $\langle t_i \rangle$ por valores de traços de itens presentes na vizinhança do item alvo. Se uma regra aplica-se a um item, então a associação de valor especificada do lado direito pode ser realizada nesse item.

Como foi mencionado na seção anterior, o método TBL gera regras com base em moldes que determinam os tipos de expressões condicionais possíveis. Nos pares (TA,$val$) de uma expressão condicional, o TA determina o item e o traço que, durante o processo de aprendizado, terá seu valor capturado em $val$ para compor a regra. Dessa forma, um molde é simplesmente uma sequência de TAs:

$$\langle t_1 \rangle \ \langle t_2 \rangle \ \langle t_3 \rangle \ \ldots \ \langle t_n \rangle$$

Nas aplicações de TBL encontradas na literatura, os TAs geralmente possuem um dos formatos (a) e (b) mostrados a seguir:

(a) ***ftr_índice***: captura o traço *ftr* de um item que se encontra deslocado, para a esquerda ou para a direita, *índice* posições em relação ao item alvo. Exemplo de TA para tal padrão seria: $word\_0$, que identifica o traço $word$ do item alvo.

(b) ***ftr [índice_inicial; índice_final]***: captura o traço *ftr* num intervalo de itens posicionados entre *índice_inicial* e *índice_final*, em relação ao item alvo. Um exemplo de TA para tal padrão seria $word[1;3]$, que captura uma determinada unidade léxica nos itens das posições +1, +2 e +3 em relação ao item alvo.

O seguinte molde de regras é formado por esses padrões de TAs:

$$tsn\_-1 \;\; tsn\_0 \;\; tpos\_0 \;\; word[1;2]$$

Se usarmos esse molde para gerar regras que corrijam o erro de classificação da preposição *em* – que está marcada como *I* e deveria estar como *O* – na seguinte sentença :

A/ART/I menina/N/I deixou/V/O a/ART/I boneca/N/I **em/PREP/I** a/ART/I cama/N/I

serão geradas as seguintes regras:

$$tsn\_-1 = I \;\; tsn\_0 = I \;\; tpos\_0 = PREP \;\; word[1;2] = a \;\; \Rightarrow \;\; tsn = O$$

$$tsn\_-1 = I \;\; tsn\_0 = I \;\; tpos\_0 = PREP \;\; word[1;2] = cama \;\; \Rightarrow \;\; tsn = O$$

cuja primeira regra deve ser lida como, "**Se** $tsn\_-1 = I$ **e** $tsn\_0 = I$ **e** $tpos\_0 = PREP$ **e** $word[1;2] = a$ **Então** $tsn\_0 = O$" (**Se** o traço $tsn$ do item anterior tiver valor $I$ **e** os traços $tsn$ e $tpos$ do item atual (alvo) forem iguais a $I$ e $PREP$, respectivamente, **e** o traço $word$ de um dos dois próximos itens for *"a"* **Então** mudar o valor do traço $tsn$ para $O$ no item alvo).

## 4. Ferramenta TBL Proposta

### 4.1. Termo Atômico com restrição: uma nova abordagem de molde de regras

Os tipos de TAs (a) e (b) mostrados na seção 3.2. são apropriados para tarefas em que o tamanho da janela de contexto – onde a informação que leva à classificação correta de um item deve ser encontrada – é bem delimitado e relativamente pequeno. Nas aplicações de TBL encontradas na literatura geralmente é utilizada uma janela de sete itens de tamanho, incluindo o item alvo, os três anteriores e os três posteriores. Esse tipo de molde não é viável para certos problemas de PLN onde a classificação depende de itens com distâncias variáveis entre si.

A identificação de SNs que incluem preposições é um desses problemas. Essa tarefa é mais complexa do que a identificação de SNs básicos, visto que envolve o problema de ligação do sintagma preposicionado, que é a questão de distinguir quando a preposição está introduzindo um complemento de um verbo (e deve ser classificada como fora do SN anterior a ela), ou quando está introduzindo o complemento de um nome (e deve ser classificada como pertencente ao SN anterior). Esse é um caso onde uma janela de contexto com sete itens de tamanho é insuficiente.

A idéia mais proeminente que surge dessa descrição do problema de identificação de SNs é habilitar a verificação de uma possível dependência entre a preposição a ser classificada e o verbo que a precede. Ou seja, gerar regras específicas para observar uma preposição e o verbo que a antecede. Mas para se fazer isso, os seguintes obstáculos relacionados aos TAs (a) e (b) devem ser superados:

(1) Quando as expressões condicionais estão sendo geradas com o uso dos TAs (a) e (b), o valor do traço indicado é sempre capturado independentemente de qualquer pré-condição em relação ao correspondente item, exceto a distância em relação ao item alvo.
(2) Uma vez que a distância exata entre a preposição e o verbo que a precede não é conhecida, não é possível a utilização do TA do tipo (a) na tentativa de capturá-los numa mesma expressão condicional. Assumindo que o verbo precedente a uma preposição é encontrado numa janela de tamanho arbitrário, usar o TA do tipo (b) provocaria a geração de diversas regras desnecessárias, provocando, dentre outros problemas, grande consumo de memória e de tempo de execução no aprendizado.

Com a expectativa de contornar essas dificuldades, propomos um novo tipo de TA, o qual denominamos de TA com restrição, que possui uma janela de contexto com tamanho variável e um teste que precede a captura do valor de um traço. O uso desse tipo de TA assume que só se deve capturar o valor de um traço $X$ de um item, se um outro traço $Y$, no mesmo item, atender a um determinado teste condicional. O teste consiste em verificar se o valor do traço $Y$ é igual a um valor predefinido. O formato do TA proposto é o seguinte: *traçoX [ind_inicial;ind_final](traçoY=valY)*.

Um exemplo de TA que segue esse padrão é: $word[-2;-8](tpos = V)$, que deve ser interpretado como "Capturar o traço $word$ do item mais próximo ao item alvo, que esteja entre o intervalo fechado -2 e -8, e cujo traço $tpos$ é igual a $V$".

Com esse tipo de TA podemos construir um molde que gera expressões condicionais que observem exatamente uma preposição e o verbo que a antecede. Tal molde teria a forma: $tsn\_0 \;\; word[0;0](tpos = PREP) \;\; word[-2;-10](tpos = V)$, e quando aplicado para gerar regras na sentença mostrada em 3.2. só geraria a seguinte regra:

$tsn\_0$ =I $\; word[0;0](tpos = PREP)$ =em $\; word[-2;-10](tpos = V)$ =esqueceu $\Rightarrow \; tsn$=O

que deve ser lida como: "**SE** no item alvo (índice 0) o traço $tsn$=I **e** os traços $tpos$=PREP **e** $word$=PREP, **e** o primeiro item no intervalo [-2;-10] que atenda ao teste $tpos$=V também atender a $word$=esqueceu **ENTÃO** mudar o valor do traço $tsn$ para $O$ no item alvo".

Para a implementação da ferramenta TBL proposta nesse trabalho foi escolhido o algoritmo *FastTBL* [Ngai and Florian ]. Optamos por essa versão do algoritmo TBL porque é bem mais rápida que a versão original proposta por Eric Brill e mantém a mesma eficácia dos resultados. Maiores detalhes com relação aos TAs com restrições podem ser encontrados em [Santos 2005].

## 5. Experimentos e resultados

### 5.1. Derivação dos corpora para treino e testes

Os corpora de treino e testes utilizados nesse estudo foram derivados do Mac-Morpho, um "corpus de 1,1 milhão de palavras retiradas a partir de 1 ano de publicação (1994) do jornal brasileiro Folha de São Paulo" [Marchi 2003], disponibilizado via web pelo projeto Lacio-Web[1], do Núcleo Interinstitucional de Lingüística Computacional (NILC)[2]. Tal corpus está etiquetado morfossintaticamente.

[1]www.nilc.icmc.usp.br/lacioweb/
[2]www.nilc.icmc.usp.br

Na derivação dos corpora usamos o *parser* PALAVRAS [Bick 2000] para obtermos a análise sintática de todos os textos do Mac-Morpho. Foi desenvolvido um programa para identificar os limites dos SNs a partir da análise sintática feita pelo PALAVRAS. Com uso dessas ferramentas construímos um corpus de treino que contém 500 mil (500k) *tokens* e um corpus de testes que contém 50k *tokens*. Outro aspecto importante é que todos os verbos existentes nos corpora foram reescritos no infinitivo. A Figura 2 mostra um exemplo do corpus de treino resultante.

```
O_ART_I time_NPROP_I está_V_O quase_ADV_O rebaixado_PCP_O para_PREP_O
a_ART_I segunda_ADJ_I divisão_N_I ._._O
```

**Figura 2. Exemplo do corpus de treino**

### 5.2. Classificação inicial

No caso da identificação de SNs, a classificação inicial consiste em atribuir uma etiqueta SN a cada item do corpus. Nos experimentos realizados nesse trabalho a classificação inicial consistiu em atribuir a cada item do corpus a etiqueta SN que foi mais freqüentemente associada à etiqueta morfossintática daquele item no corpus de treino. A única exceção foi no caso das preposições, cuja etiquetagem inicial foi realizada tomando-se em consideração a unidade léxica e não a etiqueta morfossintática.

### 5.3. Moldes de regras

Para a identificação de SNs básicos do inglês com TBL, [Ramshaw and Marcus 1995] utilizaram um conjunto contendo 100 moldes de regras formados por termos atômicos que referenciam combinações de etiquetas SN com unidades léxicas e combinações de etiquetas SN com etiquetas morfossintáticas. As regras geradas por esse conjunto de moldes possuem uma janela de contexto de no máximo sete itens.

Para a identificação de SNs do português foram realizados experimentos com os seguintes conjuntos de moldes:

(C1) o conjunto de moldes de regras de [Ramshaw and Marcus 1995];

(C2) uma versão estendida de C1, onde todos os TAs que tinham intervalo [-1,-3] e [1,3], foram estendidos para [-1,-6] e [1,6], respectivamente; além disso foram incluídos mais 24 moldes que fazem referência aos traços $tsn$ e $tpos$ num contexto local e ao traço $word$ num contexto de até 8 itens para a direita e para a esquerda ;

(C3) um conjunto contendo os 80 moldes de regras de [Ramshaw and Marcus 1995] que não possuem TAs do tipo *ftr[índice_inicial;índice_final]*, juntamente com 6 moldes, contendo TAs com restrição, desenvolvidos especificamente para classificação de preposições, como mostrado na Figura 3. Esses 6 moldes de regras foram projetados para checarem alguns itens que possam contribuir com informações para a resolução da ligação de sintagmas preposicionados, usando uma janela de contexto de até vinte itens para a esquerda, na tentativa de ligar a preposição ao verbo que a precede ou à primeira unidade léxica que está classificada como fora de um SN (etiquetada com "O");

(C4) um conjunto de moldes de regras derivado de C3, onde os 6 moldes que usam TAs com restrição foram remodelados usando apenas TAs tradicionais.

Os principais objetivos do uso dos conjuntos C1, C2, C3 e C4 foram:

```
1.  tsn_-1  tpos[0;0](tpos=PREP)  word[-1;-20](tsn=O)
2.  tsn_-1  word[0;0](tpos=PREP)  word[-1;-20](tsn=O)
3.  tsn_-1  tsn_1  tpos[0;0](tpos=PREP)  word[-1;-20](tpos=V)
4.  tsn_-1  tsn_1  word[0;0](tpos=PREP)  word[-1;-20](tpos=V)
5.  tsn_-1  tsn_1  tsn_0  tpos[0;0](tpos=PREP)  word[-2;-20](tpos=V)
6.  tsn_-1  tsn_1  tsn_0  word[0;0](tpos=PREP)  word[-2;-20](tpos=V)
```

**Figura 3. Moldes de regras que contêm TAs com restrição destinados à classificação de preposições**

- verificar o desempenho da identificação de SNs do português usando um contexto local (conjunto C1) e um contexto ampliado (conjunto C2) usando apenas TAs tradicionais (TAs do tipo (a) e (b), vide seção 3.2.);
- verificar se o uso dos TAs com restrição (conjunto C3), propostos nesse trabalho, realmente pode melhorar a classificação específica das preposições, com relação aos outros tipos de TAs, na identificação de SNs do português;
- verificar as vantagens do uso dos TAs com restrição em relação aos TAs tradicionais, com a comparação dos resultados obtidos com o uso de C3 e C4.

A busca por um método de melhorar a classificação das preposições deve-se principalmente ao fato de que, nos resultados da classificação inicial e até mesmo da aplicação das regras aprendidas com o conjunto C1, os erros de preposições representaram, em média, mais de 45% dos erros totais. Essa grande proporção deve-se ao fato já citado de que a classificação das preposições exige um contexto mais abrangente.

### 5.4. Resultados

Os resultados são reportados em termos de *precisão geral*(*accuracy*) da classificação item-a-item, precisão da identificação de SNs (percentual de SNs identificados que estavam corretos), abrangência da identificação de SNs (percentual de SNs existentes no corpus de teste e que foram identificados corretamente) e medida F ($F_{\beta=1}$, média harmônica entre precisão e abrangência). Os resultados da aplicação ao corpus de teste, das regras aprendidas usando os conjuntos de moldes C1, C2, C3 e C4, são mostrados na Tabela 1.

**Tabela 1. Resultados da aplicação das regras aprendidas usando o corpus de 500k e os diferentes conjuntos de moldes de regras**

| Medida | Conjunto de Moldes C1 | Conjunto de Moldes C2 | Conjunto de Moldes C3 | Conjunto de Moldes C4 |
|---|---|---|---|---|
| Precisão Geral | 97,2% | 97,4% | **97,4%** | 97,2% |
| Abrangência | 84,6% | 85,5% | **85,9%** | 84,3% |
| Precisão | 84,8% | 85,6% | **86,6%** | 84,7% |
| $F_{\beta=1}$ | 84,7% | 85,6% | **86,2%** | 84,5% |
| Erros de class. de preposições | 635 | 592 | **521** | 632 |

Observando a Tabela 1, pode-se verificar que os melhores resultados foram conseguidos com o uso do conjunto de moldes de regras contendo TAs com restrição (conjunto C3). No treinamento com o conjunto de moldes C3, houve um aumento de 1,7% de $F_{\beta=1}$ em relação ao experimento usando o conjunto C4, e a redução de erros de preposições foi de 18%, 12% e 17,6% em relação aos conjuntos de moldes C1, C2 e C4, respectivamente.

Outras vantagens dos TAs com restrição puderam ser observadas nos experimentos usando os conjuntos C3 e C4. Os resultados confirmaram as nossas afirmações sobre os problemas do uso dos TAs tradicionais para a construção de regras que envolvem itens com contexto variável e extenso – vide seção 4.1. – e comprovam que os TAs com restrição podem solucionar tais problemas. Usando TAs com restrição o tempo de treinamento foi reduzido em 4,25 vezes. A quantidade de regras geradas a partir dos moldes que continham TAs com restrição foi bem menor que quantidade de regras geradas a partir dos mesmos moldes que continham TAs sem restrição. Essa redução de tempo de treinamento e número de regras aprendidas deve-se ao fato de que os 6 TAs com restrição do conjunto C3 geram regras apenas para a correção da classificação de preposições – induzindo a um menor número de regras candidatas – enquanto que os respectivos 6 TAs do conjunto C4 geram regras para correção de qualquer tipo de erro de classificação – induzindo a um grande número de regras candidatas.

## 6. Conclusões

Nesse trabalho, o problema da identificação de SNs do português foi abordado com o uso do algoritmo TBL. Um novo tipo de Termo Atômico – o TA com restrição – foi proposto para diminuir os erros de classificação de preposições.

Os TAs com restrição mostraram-se um mecanismo viável para tratar problemas de PLN que requerem contextos de tamanhos variáveis ou simplesmente extensos. Eles também podem ser usados para a geração de regras que corrigem erros específicos. Nos experimentos mostrados, o conjunto de moldes de regras usando TAs com restrição foi mais eficaz do que os conjuntos de moldes que usavam apenas TAs tradicionais.

## Referências


Bick, E. (2000). *The Parsing System Palavras: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. PhD thesis, Aarhus University.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4).

Lobato, L. M. P. (1986). *Sintaxe Gerativa do português: da teoria padrão à teoria da regência e ligação*. Editora Vigília, Belo Horizonte.

Marchi, A. R. (2003). Projeto lacio-web: Desafios na construção de um corpus de 1,1 milhão de palavras de textos jornalísticos em português do brasil. In *51º Seminário do Grupo de Estudos Lingüísticos do Estado de São Paulo*, São Paulo, Brasil.

Ngai, G. and Florian, R. Transformation-based learning in the fast lane. In *Proceedings of North American Chapter of the Association for Computational Linguistics*.

Perini, M. A. (2003). *Gramática Descritiva do Português*. Editora Ática, São Paulo, 4 edition.

Ramshaw, L. and Marcus, M. (1995). Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, New Jersey, USA. ACL.

Santos, C. N. (2005). Aprendizado de máquina na identificação de sintagmas nominais: o caso do português brasileiro. Master's thesis, IME, Rio de Janeiro - RJ.