Anais do II CONEL - Congresso Nacional de Estudos Linguísticos
Programa de Pós-Graduação em Linguística - Universidade Federal do Espírito Santo
Vitória (ES) • 2014 • ISSN 2237-9789

SISTEMA IDENTIFICADOR DE SINTAGMAS VERBAIS DO PB

Henrique Chaves[1], Heliana Mello[2]
[1] Universidade Federal de Minas Gerais
Belo Horizonte, Minas Gerais, 31270-901, Brasil.
[2] Universidade Federal de Minas Gerais
Belo Horizonte, Minas Gerais, 31270-901, Brasil.
hnrqchvs@outlook.com, heliana.mello@gmail.com

RESUMO
O objetivo deste trabalho foi construir um sistema para identificar e recuperar, automaticamente, sintagmas verbais contidos em um corpus representativo do Português Brasileiro (o C-Oral Brasil - http://www.c-oral-brasil.org/). O sistema a ser desenvolvido deve ser capaz de processar documentos, analisar sintaticamente as sentenças a fim de obter a sua composição de acordo com a gramática sintagmática do PB, bem como identificar todos os sintagmas verbais da sentença, recuperar os elementos e informações morfológicas que os compõem. A implementação foi feita utilizando-se Python, o kit de desenvolvimento para aplicações de Processamento de Linguagem Natural (PLN), o *Natural Language Toolkit* (NLTK).

0 INTRODUÇÃO

Este trabalho tematiza o aprendizado de informações lexicais pelo computador a partir de um *corpus* representativo do Português Brasileiro (o C-Oral Brasil disponível em http://www.c-oral-brasil.org/ [4]) e a utilização desses dados para uma análise sintática computacional.

Diante das exigências relativas ao processamento da linguagem, desenvolvemos uma ferramenta de análise sintática automática implementada em Python, o SISVPB (Sistema Identificador de Sintagmas Verbais do Português Brasileiro), a fim de extrair Sintagmas Verbais do PB e implementarmos protótipos capazes de modelar a estrutura sintagmática de uma língua natural.

Para o desenvolvimento do trabalho encontramos uma situação ideal na qual o referido *corpus*, já se encontrava etiquetado morfologicamente pelo software *Palavras*, permitindo-nos alcançar resultados melhores. Esse facilitador se deve ao fato de que uma vez etiquetados, o corpus pode ser utilizado para a extração de informações lexicais.

Antes de avançarmos nosso artigo, falaremos um pouco acerca do *Natural Language Toolkit* (NLTK). Trata-se de uma biblioteca desenvolvida em Python que funciona como uma caixa de ferramentas computacionais voltadas tanto para o Processamento de Língua Natural quanto para a análise computacional da linguagem. Destacamos ainda que o NLTK disponibiliza técnicas de aprendizado de máquina para a construção de etiquetadores morfossintáticos. Como mostraremos neste trabalho, as informações definidas por tais ferramentas podem ser integradas ao componente sintático de uma gramática para a construção do nosso Sistema.

Dessa forma, na elaboração do SISVPB para que ele tivesse uma ampla cobertura no NLTK, o componente de maior volume informacional seria, evidentemente, o léxico. Porém, modelarmos manualmente o léxico demandaria muito tempo e nos exigiria uma sofisticada engenharia de conhecimento linguístico. O NLTK, por sua vez nos disponibiliza técnicas de aprendizado para a construção de etiquetadores morfossintáticos. Como mostraremos neste trabalho, tais informações podem ser integradas ao componente sintático de uma gramática.

Neste trabalho, apresentamos uma proposta de integração de etiquetadores na análise sintática automática em moldes gerativos. Para essa tarefa, desenvolvemos o SISVPB, essa ferramenta permite construir um sistema com base em uma gramática de estrutura sintagmática em um etiquetador morfossintático especificado pelo usuário. Desse modo, podemos reutilizar, no processamento computacional da sintaxe, ferramentas e recursos livremente disponíveis no âmbito da linguística computacional e da linguística de *corpus*.

Este trabalho descreve como o SISVPB utiliza o *Palavras* para gerar, por meio de etiquetadores morfossintáticos, as informações lexicais de que um analisador sintático necessita para construir representações arbóreas para sentenças consideradas gramaticais com base nas regras de estruturação sintagmática fornecidas pelo usuário, bem como extrair os Sintagmas Verbais da mesma. Utilizando gramáticas simples que modelam apenas uma gama pequena fenômenos sintáticos do PB, o sistema produz resultados promissores, em se tratando de um protótipo, na análise de sentenças de mediana complexidade sintática, permitindo-nos extrair sintagmas Verbais do *corpus*.

1 AQUISIÇÃO LEXICAL PELO SISVPB

Nesta seção, apresentaremos brevemente a noção de aquisição lexical e argumentaremos a favor de que etiquetadores morfossintáticos, treinados em corpora por meio de técnicas de aprendizagem de máquina, podem fornecer informações para a solução do problema da modelação do léxico em sistemas de *parsing*.

Conforme Lemnitzer e Wagner (2004, p. 245), [5] consideramos aquisição lexical (*lexical acquisition*) uma tecnologia do texto, em que a extração de descrições lexicais a partir de textos é o foco.

Algumas experiências descritas na literatura (BRANCO; SILVA, 2004) [3], nos mostram que etiquetadores morfossintáticos baseados em arquiteturas robustas e treinados num *corpus* suficientemente representativo, não têm dificuldade em lidar com múltiplas categorizações sintáticas de palavras. Dada grande disponibilidade de *corpus* anotados morfossintaticamente, a construção de etiquetadores morfossintáticos tornou-se, nos últimos anos, uma tarefa relativamente simples, devido, sobretudo a ferramentas de aprendizado de máquina já existentes como o MXPOST (RATNAPARKHI, 1996) [6], o HunPos (HALÁCSY; KORNAI; ORAVECZ, 2007) [4], entre vários outras, além das que integram o NLTK (BIRD; KLEIN; LOPER, 2009; 2011) [1] [2].

Dessa forma, na construção do léxico, partimos de uma lista de extensa de palavras agrupadas em categorias lexicais (verbos, adjetivos, etc.). Vale destacar que o *Palavras* não tem dificuldade em lidar com as múltiplas categorizações sintáticas de palavras e dada a disponibilidade de um *corpus* etiquetado, torna a construção de etiquetadores morfossintáticos tarefa quase trivial, graças, sobretudo as ferramentas de aprendizado de máquina disponíveis.

2 DESENVOLVIMENTO

O NLTK oferece diversos formalismos para a construção de analisadores sintáticos. Na fase inicial de nosso analisador, utilizamos o formalismo da gramática livre de contexto (doravante CFG, do inglês *context-free grammar*), comumente utilizado na linguística computacional para a implementação de protótipos capazes de modelar a estrutura sintagmática de uma língua natural.

Uma gramática CFG é como o esqueleto a partir do qual podemos desenvolver gramáticas mais sofisticadas, capazes de modelar, de forma elegante e computacionalmente eficiente, fenômenos sintáticos mais complexos como, por exemplo, a concordância.

Simplificando, podemos definir uma CFG como uma gramática constituída de regras de reescrita (*rewriting rules*) de um dos seguintes tipos:

• A -> B, em que A é um símbolo não terminal e B, um símbolo não terminal ou uma concatenação de símbolos não terminais.

• C -> d, em que C é um símbolo não terminal e d, um símbolo terminal.

Para a PLN, os símbolos terminais representam os itens do léxico, ao passo que os símbolos não terminais pré-terminais constituem categorias lexicais e os demais não terminais, categorias sintagmáticas.

(1) Minigramática I

```
S -> NP VP
NP -> Det N
VP -> V
VP -> V PP
PP -> P NP
```

As regras acima descrevem a estrutura sintagmática (onde NP, PP e VP são categorias sintagmáticas, Det, N, P e V, categorias lexicais). As entradas lexicais são modeladas pelos etiquetadores morfossintáticos já pré-definidos pelo usuário.

(2)
```
>>> import nltk
>>> gram = ”””
S -> NP VP
NP -> Det N
VP -> V
VP -> V PP
PP -> P NP
“””
>>>gramatica = nltk.parse_gram(gram)
>>>type(gramatica)
<class ‘nltk.Grammar.ContextFreeGrammar’>
>>>gramatica
<Grammar with 13 productions>
>>>analisador
>>>nltk.parse.chart.ChartPaser object at 0x2689750
>>>arvores=analisador.nbest_parse(sentence).split(
))
>>>nltk.draw.draw_trees(*arvores)
```

Para a análise das *strings* de uma determinada língua livre de contexto (o tipo de língua na Hierarquia de Chomsky gerada por uma CFG), o NLTK oferece classes que implementam alguns dos algoritmos mais conhecidos.

Para construir o SISVPB, primeiramente importamos a biblioteca NLTK. Em seguida, atribuímos a uma variável gram uma cadeia com as produções da gramática. A aplicação da função nltk.parse_gram() sobre essa cadeia resulta em objeto da classe nltk.Grammar.ContextFreeGrammar, dado, por sua vez, como argumento do método de inicialização da classe nltk.ChartParser. Esse último é então aplicado sobre uma sentença e a representação arbórea produzida é exibida numa janela adicional.

Na fase inicial de desenvolvimento de nossa gramática, restringimo-nos a sentenças declarativas sem estruturas de coordenação, mas incluindo construções com *que*. Uma limitação desse modelo inicial é que, da teoria X-barra, implementamos diferentes níveis de projeção apenas para algumas categorias, sem adotar uma ramificação estritamente binária. Outros princípios da teoria X-barra como a endocentricidade e a maximalidade também não foram considerados de forma estrita.

No formalismo da CFG do NLTK, o único recurso de que dispomos para simplificar a elaboração das regras da gramática é o operador de disjunção lógica “|”, daí conseguimos uma versão mais compacta da nossa minigramática. Utilizamos ainda os operadores {} que permitem simplificar ainda mais as regras de uma CFG, ao mesmo tempo em que tornam o formalismo mais expressivo.

Os símbolos “*” e “+” funcionam como multiplicadores que assumem qualquer número natural como valor, começando, respectivamente, em 0 e 1. Dessa forma, uma regra como A -> B C*, onde C representa uma categoria individual ou uma disjunção como {E/F}, equivale a A -> B, A -> B C, A -> B C C, A -> B C C C etc. A regra A -> B C+ expande-se de forma análoga, com exceção da regra A -> B, pois a categoria C deve ocorrer pelo menos uma vez. Infelizmente, o NLTK não suporta regras no formato de (33). A fim de poder usufruir das vantagens desses recursos notacionais, construímos módulo em Python, denominado MyGrammar, que converte essa notação na do NLTK.

3 FUNCIONAMENTO E AVALIAÇÃO DO SISVPB

Esta seção explica o funcionamento do SISVPB apresenta os resultados de uma avaliação preliminar dos nossos protótipos. Como vimos, o NLTK não suporta o formato compacto de descrição da sintaxe de uma língua natural por meio da CFG, exemplificado na seção anterior. Primeiramente, portanto convertemos o formato descrito em outro de modo automático por meio do módulo MyGrammar. Em seguida, para compilação do Sistema, utilizamos o módulo SISVPB que construímos especialmente para possibilitar a integração da etiquetagem morfossintática como componente lexical da análise sintática automática.

(3) >>> import MyGrammar
>>>MyGrammar.escreve_gramatica("MyGrammar")

NLTK Grammar successfully processed and saved in MyGrammar.nltk

A gramática no formato do NLTK gerada pelo MyGrammar, denominada MyGrammar.nltk, é dada então como entrada para o SISVPB. Tal sistema dispõe de uma função por meio da qual podemos construir um *parser* com base em uma gramática CFG no formato do NLTK e do vocabulário que é extraído de um conjunto de sentenças, que são etiquetadas pelo *Palavras*, acionado automaticamente pelo SISVPB. Após aplicação do *parser* às sentenças do arquivo texto analizado, é exibida para cada uma a quantidade de análises geradas em formato de árvores. Uma versão desse arquivo com as sentenças anotadas (nomeado sentencas.nltk.txt) é criada no diretório de trabalho. Para produzir as árvores basta executarmos o comando (4):

(4) >>>from Palavras import AnotaCorpus
>>> t=AnotaCorpus.TokPort.tokenize
>>>a=sisvpb.analisaSentenca(t(s1),"gramaticas/MyGrammar.nltk")
>>> sisvpb.exibeArvores(a)

Feito isso, nossa tarefa de extrair Sintagmas Verbais tornou-se relativamente simples. Basta executarmos o método próprio do módulo e extrairmos o sintagma desejado:

(5) >>>sisvpb.exibeSV(a).

4 CONCLUSÕES

Neste artigo, mostramos como utilizar, na análise sintática computacional, informações lexicais extraídas de um *corpus* do PB, o c-Oral Brasil, de modo a poder prescindir tanto da elaboração de um analisador morfológico quanto da compilação de centenas de milhares de entradas lexicais.

Para tanto, desenvolvemos o SISVPB, módulo em Python capaz de integrar as informações oriundas de etiquetadores morfossintáticos de diversas arquiteturas. Com isso, visamos a contribuir para preencher lacuna no repertório de ferramentas para o processamento computacional do PB.

Optamos também por aproveitar os resultados das pesquisas em aprendizado de máquina, que permitem, hoje, construir etiquetadores eficientes com pouco ou nenhum esforço, a partir de um corpus anotado morfossintaticamente.

O SISVPB, devido a existência de tais ferramentas é capaz de lidar de forma bastante eficiente com as dificuldades lexicais típicas de processamento de textos.

No atual estágio de desenvolvimento, o SISVPB trata-se de um protótipo com resultados interessantes, haja vista a relativamente pequena cobertura das gramáticas que avaliamos neste trabalho bem como a necessidade de melhorar a correção dos erros mais frequentes na etiquetagem dos textos.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] BIRD, S.; KLEIN, E.; LOPER, E. *Natural language processing with Python*: analyzing text with the Natural Language Toolkit. Sebastopol: O'Reilly, 2009. 502 p.

[2] BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Toolkit*. [s.l]: [s.n.], 2011. Disponível em: <http://www.nltk.org>. Acesso em: 24 jan. 2011.

[3] BRANCO, A.; SILVA, J. 2004. Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese. In: LINO, M. T. et al. (Ed.). INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION, n. 4, 2004, Lisboa. *Proceedings*... Paris: ELRA, 2004. p. 507-510.

[4] Raso, Tommaso & Mello, Heliana. C-ORAL-BRASIL I: *Corpus de referência do português brasileiro falado informal*. Belo Horizonte: Editora UFMG. 2012.

[5] HALÁCSY, P.; KORNAI, A.; ORAVECZ, C. HunPos: an open source trigram tagger. ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, n. 45, 2007, Praga. *Proceedings*... Stroudsburg: Association for Computational Linguistics, 2007. p. 209-212.

[6] LEMNITZER, L.; WAGNER, A. Akquisition lexikalischen Wissens. In: H. LOBIN; LEMNITZER (Ed.). *Texttechnologie*: Perspektiven und Anwendungen. Tübingen, Stauffenburg, 2004. p. 245-266.

[7] RATNAPARKHI, A. A Maximum Entropy Model for Part-Of-Speech Tagging. EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING, 1996, Philadelphia, Pennsylvania. *Proceedings*... Pennsylvania: University of Pennsylvania, 1996. p. 133-142.
Disponível em:
<http://acl.ldc.upenn.edu/W/W96/W96- 0213. pdf>. Acesso em:<2. Jun. 2013>