# Parsing Probabilístico para o Português do Brasil

Andréia Gentil Bonfante[1] e Maria das Graças Volpe Nunes[2]

[1] Instituto de Ciências Matemáticas e de Computação,ICMC, USP São Carlos / Faculdades COC, Ribeirão Preto, SP
andreia@nilc.icmc.sc.usp.br
http://www.nilc.icmc.sc.usp.br
[2] Instituto de Ciências Matemáticas e de Computação,ICMC, USP São Carlos
mdgvnune@.icmc.sc.usp.br
http://www.icmc.sc.usp.br/∼mdgvnune

**Resumo** Este projeto de doutorado, em andamento, investiga um parser empírico probabilístico para o português brasileiro, cujo modelo de linguagem seguido é baseado num conceito head-centered, onde o núcleo é o elemento chave e direcionador, com os demais elementos como seus modificadores. Como treebank alimentadora do processo, utilizamos um conjunto de sentenças obtidas do Corpus Nilc (www.nilc.icmc.sc.usp.br) anotadas sintaticamente com o parser de E. Bick [1], uma anotação híbrida que anota a estrutura em si, mas também as funções sintáticas de cada sintagma. O modelo probabilístico usado segue os padrões de Collins [10], cuja eficiência chega a 88% para sentenças na língua inglesa. O parser segue os moldes de um chart-parser tradicional, recuperando os sintagmas formadores da sentença num processo bottom-up, com a diferença que, ao invés de se basear em regras para a junção das palavras em sintagmas, o modelo aqui descrito utiliza informações probabilísticas sobre a possibilidade de duas palavras se juntarem num mesmo sintagma, uma delas sendo núcleo e outra, modificador. Duas fases distintas são destacadas: a fase de obtenção dos sintagmas nominais e a fase de obtenção dos demais sintagmas. Até a presente data, estamos no processo de finalização da anotação das sentenças do corpus (conjunto de sentenças) de teste com o parser implementado, para posteriormente avaliá-las de acordo com medidas como precision e recall, onde os sintagmas formados são avaliados. A avaliação segue o esquema ten-fold crossvalidation, com cada fold contendo 90.000 sentenças para treinamento e 10.000 para teste.

**Palavras chaves:** Processamento de Língua Natural, Parsing Empírico

## 1 Introdução

Seja para o processamento das línguas naturais, ou para alimentar um mecanismo de busca, um dos requisitos fundamentais para o bom funcionamento de

sistemas de tal natureza é a eficiência conseguida na recuperação da estrutura sintática das sentenças a serem tratadas. Muito esforço tem sido empregado na construção de analisadores sintáticos (parsers) para o reconhecimento dessas estruturas. A dificuldade na especificação de gramáticas com poder de descrição considerável para ser usada por um parser abriu caminho para a pesquisa em um segmento alternativo chamado empírico. Nessa linha, um conjunto de sentenças anotadas sintaticamente (treebank) é fonte para a obtenção de modelos do conjunto de treinamento, num processo de aprendizado, que servirá de base para uma futura anotação de uma sentença ainda não vista.

As pesquisas com métodos empíricos estendem-se englobando três abordagens principais das áreas de aprendizado: o aprendizado de máquina simbólico, o aprendizado neural conexionista e o aprendizado estatístico.

As abordagens simbólicas para o aprendizado incorporam uma grande variedade de técnicas de aprendizado de máquina. Esses sistemas têm por característica a aquisição da informação de forma automática, processo chamado de indução, a partir de uma base de exemplos, e a representação do conhecimento inferido em formas que podem ser variadas. Para aplicações de linguagem, existem sistemas empregando várias dessas formas de representação, como árvores de decisão [16], regras de transformação ([3];[4]) e programação lógica indutiva (ILP) ([24];[19]; [14]).

Já os modelos conexionistas, até então com resultados não satisfatórios quando aplicados a problemas de língua, contra-atacam na década de 90 com novas propostas, fortemente representadas pelas redes parcialmente recorrentes de Elman ([11]; [22];[18];[13]). As redes recorrentes resolvem o problema do tamanho variado das sentenças, uma vez que possuem entrada incremental. Conseguem também generalizar sobre as posições estruturais das palavras na sentença. Há ainda outras arquiteturas neurais, extensões naturais das recorrentes, como a proposta em [23] e em [15]. Os autores propõem uma arquitetura que suporta computação simbólica dentro de uma arquitetura neural. Tal arquitetura é capaz de juntar habilidades necessárias para a recuperação das características da língua, como a combinação de várias fontes de restrição e a habilidade de manipular dinamicamente as representações complexas. Apesar de algumas limitações, o sistema proposto tem se mostrado de grande eficiência.

A aplicação dos modelos estatísticos para problemas de língua natural, por outro lado, envolve o emprego de técnicas como n-gram, Hidden Markov Models (HMMs), Gramáticas Livres de Contexto Probabilísticas ([5],[6]) ou Árvores de Decisão Probabilísticas [17]. Dentre as primeiras tentativas de usar essa metodologia num processo de parsing destacam-se as gramáticas probabilísticas de Charniak, em [5]. Tais gramáticas diferem das tradicionais no sentido de que, a cada regra gramatical formada é associada uma probabilidade de ocorrência, probabilidade essa obtida a partir das observações no conjunto de treinamento. Com um resultado de aproximadamente 75%, o parser proposto por Charniak consegue extrair, num processo de aprendizado, todas as informações que precisará para analisar as sentenças. O modelo proposto por Charniak tem um desempenho não tão satisfatório por não trabalhar com itens lexicais. Propostas

com desempenho melhor ([2];[7];[9];[10];[17]; [21]) usam modelos lexicalizados, condicionando as probabilidades ao conteúdo lexical das sentenças que são analisadas. Além disso, são parsers cuja noção principal usada é a de núcleo, na qual as probabilidades dos descendentes de um constituinte/sintagma são condicionadas ao núcleo lexical do mesmo.

Este trabalho de doutorado, em andamento, visa investigar o comportamento de analisadores sintáticos, implementados seguindo cada uma das três linhas de abordagens descritas acima, quando aplicados a sentenças para a língua portuguesa do Brasil, mais complexa que a língua inglesa, e usando um conjunto de etiquetas sintáticas maiores. Para isso, utilizamos uma treebank, obtida a partir do Corpus Nilc (www.nilc.icmc.sc.usp.br), anotada sintaticamente com o parser de Bick [1]. Descrevemos aqui, uma experiência na implementação de um parser probabilístico, seguindo modelo de Collins em [10]. O modelo é derivado a partir da noção de núcleos lexicais, onde, para cada regra observada no conjunto de treinamento, as palavras que não são núcleo são chamadas de modicadores, exercendo inflência sobre ele. A formação da estrutura sintática de uma sentença se dá através de um processo bottom-up, onde a probabilidade na qual um núcleo e um modificador se juntam para formar um sintagma é a que comanda o processo.

## 2 Trabalho Anterior

Collins, em sua tese [10] propõe um modelo gerativo para implementar seu parser para a língua inglesa. Esse modelo gerativo maximiza a probabilidade de uma sentença S ter uma árvore sintática T através da atribuição de probabilidades($P(T,S)$) a uma derivação top-down da árvore para aquela sentença. Usando uma gramática probabilística livre de contexto (PCFG), com palavras $w$ e anotações morfos-sintáticas (POS tags) $t$ ligadas a não-terminais $X(x =< w,t >)$ na árvore, cada regra tem a forma

$$P(h) \rightarrow L_n(l_n)...L_1(l_1)H(h)R_1(r_1)...R_m(r_m) \quad (1)$$

onde $H$ é a anotação sintática do núcleo da regra, do filho de $P$, que herda $h$ de seu pai; $L1...Ln$ e $R1..Rm$ são modificadores à esquerda e à direita.

A geração do lado direito de uma regra ($RHS$), dado o lado esquerdo ($LHS$), é feita em três passos, primeiro gerando o núcleo, então fazendo suposições de independência para os modificadores do lado esquerdo e do lado direito, que são gerados por um processo de Markov de ordem 0:

- Gera a anotação sintática do núcleo do sintagma, com probabilidade $P_h(H|P,h)$
- Gera modificadores à direita do núcleo com probabilidade

$$\prod_{i=1...m+1} P_R(R_i(r_i)|P,h,H) \quad (2)$$

onde $R_{m+1}(r_{m+1})$ é definido como STOP - o símbolo STOP é adicionado ao vocabulário de não-terminais, e o modelo para de gerar modificadores à direita quando este é gerado.

– Gera modificadores à esquerda com probabilidade

$$\prod_{i=1...n+1} P_L(L_i(l_i)|P,h,H) \tag{3}$$

onde $L_{n+1}(l_{n+1}) = STOP$

Esse tipo de decomposição ajuda a evitar o enorme número de regras em potencial e o problema dos dados esparsos gerados pela estimativa direta de $P(RHS|LHS)$.

### 2.1 Tratamento de Dados Esparsos (palavras não vistas)

Há vários níveis de back-off para cada tipo de parâmetro no modelo. Os modificadores são decompostos e tratados separadamente. Por exemplo, $P_L(L_i(lw_i, lt_i)|P,H,w,t)$ é decomposto no produto $P_{L1}(L_i(lt_i)|P,H,w,t) \times P_{L2}(L_i(lw_i)|L_i(lt_i),P,H,w,t)$, onde $lw_i$ e $lt_i$ são a palavra a tag gerada com o não-terminal $L_i$. Nesse caso, a estimativa final é:

$$e = \lambda_1 e_1 + (1-\lambda_1)(\lambda_2 e_2 + (1-\lambda_2)e_3) \tag{4}$$

onde $e_1$ (por exemplo, $P_H(H|P,w,t)$), $e_2$ (por exemplo, $P_H(H|P,t)$) e $e_3$ (por exemplo, $P_H(H|P)$) são estimativas maximum likelihood, com o contexto em diferentes níveis (três, no caso exemplificado). Todas as palavras que ocorrem menos que 5 vezes nos dados de treinamento, e palavras nos dados de teste que nunca foram vistas durante o treinamento, são substituídas pelo token "uknown". Isso permite que o modelo lide com as estatísticas para palavras raras ou novas palavras de uma forma robusta.

### 2.2 Comentários

O modelo proposto por Collins é do tipo history-based, no qual uma árvore sintática é representada como uma seqüência de decisões, decisões estas tomadas a partir de uma derivação top-down, centrada no núcleo da árvore sintática. A representação da árvore desse modo permite que sejam feitas suposições independentes, que levam a parâmetros condicionados a núcleos lexicais, como projeção do núcleo, subcategorização, colocação de complemento/adjunto, dependência, distância, etc.

A proposta estende as gramáticas probabilísticas para gramáticas lexicalizadas, inclusive incorporando a anotação morfossintática no próprio modelo. Os modelos podem condicionar a qualquer estrutura (previamente gerada).

O modelo tem uma precision/recall de constituinte de 88,3/88% sobre [8], [6], [17], [20] e [12]. Usando tal tipo de probabilidade conjunta, o modelo também pode ser usado para reconhecimento de voz ou tradução.

## 3 Formação da Treebank

A anotação da treebank usada no experimento é proposta por Bick em [1]. Trata-se de uma anotação baseada num paradigma de gramática de dependência, onde morfologia, sintaxe e semântica são tratadas. A sintaxe dependência "flat" (plana) é enriquecida com a adição de marcadores de direção (> e <) e etiquetas de forma e função a subcláusulas, o que permite a transformação para árvores de constituintes. Com alguns scrips em Perl, os grupos e as fronteiras da descrição flat são identificados, os constituintes são delimitados, e a parentização das fronteiras dos constituintes são marcados em uma forma mais complexa (np - sintagma nominal, pp - sintagma preposicional, etc), e finalmente, a cada constituinte é atribuída uma etiqueta de função derivada de uma etiqueta sintática de seu núcleo. A tabela 1 mostra um exemplo de um sistema de anotação para a sentença "Erros abalam credibilidade da imprensa".

**Tabela 1.** Um exemplo do sistema de anotação de Bick [1]

| |
|---|
| $SUBJ : n(MP)Erros$ |
| $P : v - fin(PR\|3P\|IND)abalam$ |
| $ACC : np$ |
| $= H : n(FS)credibilidade$ |
| $= N <: pp$ |
| $== H : prp(< sam- >)de$ |
| $== P <: np$ |
| $===> N : art(< -sam > FS)a$ |
| $=== H : n(FS)imprensa$ |

onde SUBJ = sujeito; n = nome; M = masculino; P = plural; $P :=$ predicado; v-fin = verbo finito; PR = tempo passado; 3P = terceira pessoa; IND = indicativo; ACC = objeto direto acusativo; np = sintagma nominal; H = núcleo; F = feminino; S = singular; N pré/pós nominal; pp = sintagma preposicional; prp = preposição; $< sam- >$ = primeira parte de uma fusão morfológica de uma palavra (de); art = artigo.

## 4 Experimentos em andamento com o modelo head-centered probabibístico

Como parte da proposta de trabalho, desenvolvemos um modelo probabilístico e um chart parser que utiliza as informações compiladas no modelo para analisar as sentenças. O modelo segue o padrão proposto por Collins em [10], no qual cada palavra que não seja núcleo é encarada como modificador de seu respectivo núcleo. Para isso, a cada regra, são identificados o núcleo e seus modificadores,

juntamente com informações sobre suas anotações sintáticas, seus pais na hierarquia, a direção do modificador com relação ao núcleo, a adjacência ou não com seu núcleo, e a possibilidade de ser gerado um STOP logo a sua esquerda ou a sua direita.

Vários scripts em Perl são usados para transformar a notação flat inicialmente obtida em regras com núcleos identificados. Para cada regra observada, vários modelos são montados. Por exemplo, dada a sentença "Ibama revela contrabando de madeira", anotada segundo notação flat de Bick:

$Ibama[Ibama]PROPM/F\ S\ @SUBJ>$
$revela[revelar]<fmc>V\ PR\ 3S\ IND\ VFIN\ @FMV$
$contrabando[contrabando]\ NMS\ @<ACC$
$de[de]\ PRP\ @N<$
$madeira[madeira]\ N\ F\ S\ @P<$

onde PROP = nome próprio; M = masculino; F = femino; S = singular; $@SUBJ$ = sujeito; V = verbo; PR = presente; 3S = 3a. pessoa singular; IND = indicativo; $@ACC$ = objeto direto; $@N$ = adjeto adnominal; @P = predicador; $@FMV$ = verbo principal; v-fin = verbo finito; N = nome; PRP = preposição;

Essa notação é transformada em regras, onde H é núcleo:

$S \rightarrow SUBJ(ibama, prop|m/fs/p)\ P(revela, v-fin|pr3sind)\ ACC(np)$
$ACC(np) \rightarrow H(contrabando, n|ms)\ N<(pp)$
$N<(pp) \rightarrow H(de, prp)\ P<(madeira, n|fs)$

Os núcleos são identificados para cada regra. Assim, núcleo de

$S = revela, v-fin$
$ACC = contrabando, n$
$N< = de, prp$

A partir daí são montados alguns modelos, dentre os principais, os que chamamos modelo Nucleo, modelo NP e modelo Total:

- Para o cálculo da probabilidade núcleo $Ph(H|P,h)$ vista acima, o modelo núcleo guarda as informações para cada regra: nucleo,tag,anotação sintática e anotação do sintática pai. Exemplo, para a sentença acima, temos:
  $contrabando, n, H, ACC$
  $de, prp, H, N<$
  $revela, v-fin, P, S$
- Tanto para os sintagmas nominais, quanto para o restante dos sintagmas são armazenadas informações: para $Pr(Ri(ri)|P,h,H)$ e $Pl(Li(li)|P,h,H)$, cada par modificador/núcleo armazenamos informações como palavra, tag, nucleo, tagnucleo, direção com relação ao núcleo, anotação sintática da palavra, an-

otação sintática do pai, anotação sintática do núcleo, adjacência com relação ao núcleo, presença de outros verbos entre o modificador e o verbo principal, e posição na regra, se está próximo a um STOP à esquerda ou à direita.

Exemplo, para Nps temos:

$ibama, prop, revela, v-fin, l, SUBJ, S, P, 1, 0, 1, 0$
$de, prp, contrabando, d, N, N<, ACC, H, 1, 0, 0, 1$
$madeira, n, de, prp, d, P<, N<, H, 1, 0, 0, 1$

Para Total, temos
$contrabando, n, revela, v-fin, d, ACC, S, P, 1, 0, 0, 1$

Assim os modelos são montados em tabelas hash para facilitar a busca, uma vez que são 90.000 sentenças que os alimentam. O chart parser passa então a considerar os modelos como parâmetros para montagem da estrutura da sentença a ser analisada.

Por exemplo, para analisar a mesma sentença acima, o parser receberia a entrada "ibama PROP revela N contrabando N de PRP madeira N", e partir daí, faz um chart combinando palavras de duas em duas, de acordo com as probabilidades de se unirem, sendo uma palavra o núcleo e a outra seu modificador. Para Nps, o parser produzirá saidas como na Figura 1 e para a sentença inteira, como na Figura 2.

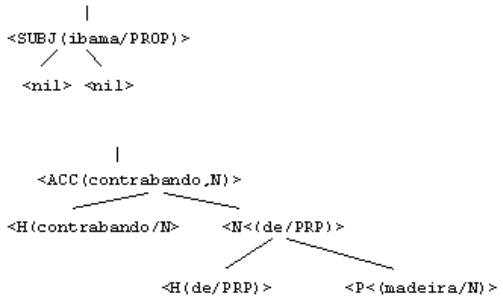


**Figura 1.** Saída produzida para os sintagmas nominais

### 4.1 Discussão e trabalhos futuros

Apesar de ainda não termos resultados concretos, alguns pontos devem ser comentados. A opção pela identificação prévia dos sintagmas nominais de cada sentença contribui com dois pontos positivos. Facilita o trabalho do parser, uma vez que, por ser um processo exaustivo, gasta tempo computacional trabalhando com possíveis combinações entre uniões improváveis. Por exemplo, modificadores dentro dos sintagmas nominais são importantes apenas localmente, e

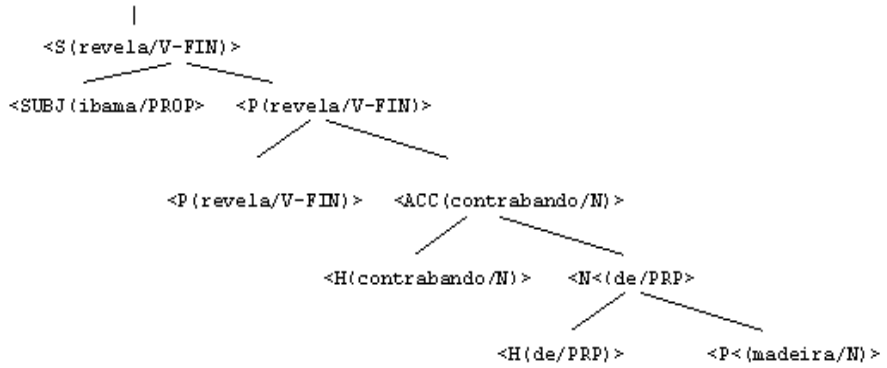


**Figura 2.** Saída produzida para a sentença

não formarão outros tipos de sintagmas. Sendo assim, podem ser descartados numa análise mais geral. Outro ponto positivo é que a identificação de sintagmas nominais são também importantes para outras aplicações como sistemas de buscas/consultas em base de dados ou via WEB. Assim podemos usar o modelo para outra aplicação além de parser.

O experimento ainda está em andamento. O processo de avaliação consistirá no esquema de ten-fold crossvalidation, com 100.000 sentenças, sendo a cada fold, separadas 90.000 sentenças para o treinamento e 10.000 sentenças para teste. Atualmente, estamos em fase de teste, onde o parser construído usa o modelo para anotar as sentenças de acordo com os padrões aprendidos, e reparação de erros de implementação/modelagem observados.

## Referências

1. E. Bick. *The Parsing System "Palavras" - Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press, 2000.
2. R. Bod. What is the minimal set of fragments that achieves maximal parser accuracy. In *Proceedings of Association for Computation Linguistics 2001*, 2001.
3. E. Brill. Automatic grammar induction and parsing free text: A transformation-based approach. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 259–265, Columbus, Ohio, 1993.
4. E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. In *Computational Linguistics*, number 21(4), pages 543–565, 1995.
5. E. Charniak. *Statistical Language Learning*. MIT Press, 1993.
6. E. Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 1997.
7. E. Charniak. A maximum-entropy-inspired parser. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, Washington, USA, 2000.

8. M. J. Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the* $34^{th}$ *Annual Meeting of the Association for Computatational Linguistics*, pages 184–191, 1996.
9. M. J. Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computatational Linguistics*, 1997.
10. M. J. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.
11. J. L. Elman. Finding structure in time. In *Cognitive Science*, number 14, pages 179–211, 1990.
12. F. J. et al. Decision tree parsing using a hidden derivation model. In *Proceedings of the 1994 Human Language Technology Workshop*, pages 272–277, 1994.
13. S. L. S. Fong and C. L. Giles. On the applicability of neural network and machine learning methodologies to natural language processing. In *Workshop on New Approaches to Learning for Natural Language Processing, IJCAI-95*, pages 1–8, Montreal, Canadá, 1996.
14. S. W. E. G.Scheler. *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*. Springer, New York, 1996.
15. J. Henderson and P. Lane. A connectionist architecture for learning to parse. In *Proceedings of the Association of Computational Linguistics - COLING-ACL*, pages 531–537, Montreal, Quebec, CA, 1998.
16. U. Hermjakob and R. Mooney. Learning parse and translation decisions from examples with rich context. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics*, pages 482–489, Somerset, N. J., 1997. Association for Computational Linguistics.
17. D. M. Magerman. Statistical decision-tree models for parsing. In *ACL Conference*, http://www-cs-students.stanford.edu/ magerman/pubs.html, June, 1995.
18. R. Miikkulainen. *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory*. MIT Press, Cambridge, MA, 1993.
19. R. Mooney. Inductive logic programming for natural language processing. In *Proceedings of the Sixth International Inductive Logic Programming Workshop on Logic Programming*, Stockholm, Sweeden, 1996. Springer Verlag.
20. A. Ratnaparkhi. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*. Brown University, Providence, Rhode Island, 1997.
21. A. Ratnaparkhi. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34:151–176, 1999.
22. R. G. Reilly and N. E. S. Eds. *Connectionist Approaches to Natural Language Processing*. Lawrence Erlbaum and Associates, Hilldale, NJ, 1992.
23. L. Shastri and V. Ajjanagadde. From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. In *Behavioral and Brain Sciences*, 1993.
24. J. M. Zelle and R. J. Mooney. Learning to parse data base queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1050–1055, Menlo Park, Calif., 1996. American Association for Artificial Intelligence.