ENSINO DE SINTAXE PORTUGUESA NA INTERNET

Eckhard Bick*

Abstract: The paper presents the VISL project; an Internet based teaching system for Portuguese grammar, based on an automatic tagger-parser for Portuguese, developed as part of a dissertation project at Århus University (Bick 1996, 2000-2). The open system relies on a lexicon of 50.000 lemas and thousands of grammatical rules to supply a complete morphological and syntactic analysis of running, natural text. The formalism used follows the Constraint Grammar tradition (CG), introduced by Fred Karlsson (1990, 1995). In spite of using a highly differentiated tag set, the parser achieves correctness rates of 99% for morphology (word class and inflexion) and 97-98% for syntax. In parallel with the open system, a data base of manually controlled sentences was created, covering various syntactic phenomena in a systematic way. In the teaching interface, users can choose between different notational filters, modelling different descriptive paradigms. Examples are word class colouring exercises or graphical syntactic trees built and tagged by the student, and controlled and commented by the computer, with form and function tags at every node. When used for grammatical corpus annotation, the parser permits complex searches, drawing on both lema, word class and syntactic function. Apart from obvious applications like lexicography and linguistics, the corpus tool can be integrated into the teaching interface to exemplify or quantify a given grammatical structure.

Key Words: NLP (Natural Language Parsing) of Portuguese, Constraint Grammar, Internet Based Distance Learning, Syntax Teaching.

Apresenta-se aqui o projeto VISL (Visual Interactive Syntax Learning), um sistema eletrônico de ensino para gramática portuguesa (http://visl.hum.sdu.dk), baseado num analisador automático (*tagger-parser*) para Português, que foi desenvolvido por mim no contexto de um projeto de doutoramento na Universidade de Århus (Bick 1996, 2000-2). O sistema aberto apóia-se num léxico de 50.000 lemas e milhares de regras gramaticais para fornecer uma análise completa, tanto morfológica como sintática, de um texto qualquer. O formalismo aí aplicado encaixa-se na tradição da Constraint Grammar (CG), introduzida por Fred Karlsson (1990, 1995). Embora usando um conjunto de etiquetas gramaticais bastante diversificado, o *parser* alcança um nível de correção de 99% em termos de morfologia (classe de palavras e flexão) e 97-98% em termos de sintaxe. Ao lado do sistema aberto, foi estabelecida uma base de orações controladas, cobrindo vários fenômenos sintáticos de uma maneira mais sistemática.

Na interface de ensino, usuários podem escolher entre vários filtros notacionais, apoiando-se em diferentes paradigmas descritivos da língua. São exemplos os exercícios nos quais colorem-se palavras para marcar sua classe, ou árvores de sintaxe gráficas construídas pelo estudante e controladas pelo computador, com etiquetas de forma e função em cada nó.

* Institute of Language and Communication, SDU – Odense University, Denmark

Quando usado para etiquetagem gramatical de *corpora*, o *parser* permite buscas complexas, juntando ao mesmo tempo palavras e lemas, classe de palavra e função sintática. Fora de aplicações óbvias como lexicografia e lingüística, essa ferramenta de trabalho com *corpus* pode integrar-se à interface de ensino, fornecendo ao estudante exemplos e quantificações de certas estruturas gramaticais.

1.2 Gramática Constritiva

Uma grande diferença entre o sistema VISL e outros sistemas de ensino de sintaxe é o fato de poder trabalhar com linguagem natural, não restrita. Isso deve-se à robustez do sistema de base, uma Gramática Constritiva (CG) para o Português.

A maioria das palavras em textos de língua natural é – quando vista isoladamente – ambígua quanto a classe de palavra, flexão, função sintática, conteúdo semântico etc. A Gramática Constritiva tenta formalizar o processo cognitivo de desambigüização em um conjunto de regras que constringem – por intermédio de condições contextuais – qual das interpretações possíveis para cada palavra será escolhida ou rejeitada. Essas regras compilam-se num *parser* "reducional" que – nos níveis de morfologia e semântica – seleciona a etiqueta certa. No nível sintático o *parser* contém regras tanto produtivas como restritivas, essas mapeando etiquetas ambíguas de função sintática, aquelas rejeitando ou selecionando etiquetas através do contexto.

Antes de embarcar no processo de desambigüização, uma CG necessita do *input* (1) de um analisador morfológico que, no caso do sistema português, trata de flexão, derivação e controle de lemas possíveis num léxico abrangente[1], de onde também tira informação secundária, de regência e de semântica, para a contextualização.

(1)
```
"<nunca>"
        "nunca" ADV
"<como>"
        "como" <rel> ADV
        "como" <interr> ADV
        "como" KS
        "como" <vt> V PR 1S VFIN
"<peixe>"
```

lineb@hum.au.dk, http://visl.hum.sdu.dk

[1] Até com 50.000 lemas o léxico português só cobre 97.6-99.7% das palavras num texto misto. Ao resto atribui-se uma análise heurística (Bick, 1998), usando afixos e formas flexionais caraterísticas, "regras" de variação ortográfica e composição de nomes próprios.

"peixe" N M S

[ADV=advérbio, KS=conjução subordinativa, V=verbo, N=substantivo, PR=presente, S=singular, P=plural, M=masculino, F=feminino, 1S=1.pessoa/singular, VFIN=verbo finito, <rel>=relativo, <interr>=interrogativo, <vt>=monotransitivo]

O conjunto ambíguo das quatro análises morfológicas da palavra *'como'* se chama uma *coorte* na terminologia da CG. Uma regra típica de desambigüização para essa coorte é a seguinte (simplificada):

(2) SELECT (VFIN) IF (NOT *-1 VFIN) (NOT *1 VFIN)

[Escolhe (SELECT) a etiqueta VFIN (verbo finito) num contexto onde não tem outro verbo finito, nem à esquerda (*-1) nem à direita (*1)]

Depois da desambigüização morfológica, o *parser* acrescenta funções sintáticas típicas para cada palavra, usando só o mínimo de condições contextuais:

(3) "<nunca>"

"nunca" ADV @ADVL

"<como>"

"como" <vt> V PR 1S VFIN @FMV

"<peixe>"

"peixe" N M S @SUBJ @ACC @SC @OC

[@ADVL=adverbial, @FMV=verbo principal finito, @SUBJ=sujeito, @ACC=objeto direto (acusativo), @SC=complemento do sujeito, @OC=complemento do objeto]

Entre as quatro etiquetas sintáticas mapeadas na palavra *'peixe'*, a de objeto direto (@ACC) pode ser selecionada *positivamente*, por intermédio de uma regra SELECT aproveitando a transitividade do verbo (<vt>), mas mais típica na tradição CG – porque é mais robusta – é a seleção *negativa*, na qual a etiqueta certa (aqui @ACC) simplesmente é a que sobrevive como última, depois de regras DISCARD rejeitarem todas as outras:

(4) DISCARD (@SUBJ) IF (0 N) (NOT *-1 V3) (NOT *1 V3)

[Rejeite (DISCARD) a etiqueta de sujeito (@SUBJ) se a palavra (0) é um substantivo (N) e não existe um verbo de 3ª. pessoa na frase]

DISCARD (@SC) IF (NOT *-1 <vK>) (NOT *1 <vK>)

[Rejeite a etiqueta de complemento do sujeito (@SC) se não existe um verbo de ligação (<vK>) na frase]

DISCARD (@OC) IF (NOT *-1 @ACC) (NOT *1 @ACC)

[Rejeite a etiqueta de complemento do objeto (@OC) se já não existe um objeto direto na frase]

Hoje existem Gramáticas Constritivas no nível morfológico para várias línguas, inglês, alemão, sueco, finlandês, entre outras. Trabalho atualmente com dinamarquês e estou também

melhorando o sistema inglês. Ano passado, adaptei a minha CG portuguesa para espanhol. Uma gramática constritiva, madura, contém tipicamente milhares de regras, 1-2000 para cada nível e destaca-se por uma grande robustez. Para o nível morfológico de inglês, por exemplo, Voutilainen (1992) relata menos de 0.3% de erros quando desambigüizando completamente 94-97% das palavras.

2 Árvores sintáticas na notação "chata" de CG

2.1 Forma e função sintáticas

Historicamente, a CG nasceu da análise morfológica, e a maioria dos sistemas baseia-se em analisadores morfológicos TWOL (Koskenniemi, 1983), as regras apoiando-se em traços morfológicos e classe de palavra. Por isso, a descrição gramatical da CG exprime-se formalmente por meio da palavra, juntando à palavra etiquetas não só simples, lexicais, mas também complexas, estruturais. Sintaxe "chata" é uma conseqüência natural disso, e também o *parser* português utiliza representações estruturais chatas. A descrição contém informação não somente sobre *função sintática* (p.ex. argumentos como @SUBJ, @ACC) como *forma sintática* (hierarquia de constituintes). Essa última exprime-se por intermédio de *marcadores de dependência* (< >), indicando a cabeça de um dado sintagma e estabelecendo implicitamente os limites sintagmáticos.

(5)	Temos	[ter] <vt> V PR 1P IND VFIN	@FMV
	em	[em] <sam-> PRP	@<ADVL
	este	[este] <-sam> <dem> DET M S	@>N
	país	[país] <top> N M S	@P<
	uns	[um] <art> DET P S	@>N
	castelos	[castelo] <hus> N M P	@<ACC
	muito	[muito] <quant> ADV	@>A
	velhos	[velho] ADJ M P	@N<

Nessa notação, cada palavra "lembra" sua relação imediata de dependência (sua cabeça sintática), e toda a estrutura sintática pode ser descrita localmente, por etiquetas baseadas em palavras. Como as partes móveis de um móbile, as palavras da frase têm que "saber" só as suas relações imediatas, quando cabeça e dependentes. Com essa informação, o móbile sempre pode ser reconstituído inequivocamente.

Para tratar de orações subordinadas, junto uma segunda etiqueta, "exterior", a uma palavra chave da oração subordinada, – ou no primeiro verbo ou no subordinador. No

exemplo abaixo, o subordinador/complementizador "que" informa-nos que a oração subordinada é finita (@#FS) e tem o papel de objeto direto (<ACC) do verbo principal.

(6)	Sabe	[saber] <vq> V PR 3S IND	@FMV
	que	[que] KS	**@#FS-<ACC @SUB**
	os	[o] <art> DET M P	@>N
	problemas	[problema] N M P	@SUBJ>
	são	[ser] <vK> V PR 3P IND	@FMV
	graves	[grave] ADJ M/F P	@<SC

2.2 Transformação da "dependência chata" em árvores sintáticas

Tendo em vista a grande popularidade da tradição gerativa no ensino lingüístico, e dada a qualidade pedagógica de uma apresentação gráfica de estruturas sintáticas, leva-se a pergunta de equivalência e transformabilidade entre as duas notações, CG "chata", de um lado, e árvores sintáticas, do outro lado. Será que com um conjunto de etiquetas bastante rico, e com marcadores de dependência em todos os níveis, uma análise feita pela CG pode ser transformada numa árvore sintática? Para comprovar essa hipótese, escrevi um programa de transformação que insere marcadores de limites de constituintes (*constituent boundaries*) na notação chata, usando regras como a do não-cruzamento de dependências do mesmo nível, *uniqueness principle* (só *um* argumento do mesmo tipo por oração, sem co-ordenação<ATENÇÃO MARCELO: coordenação?). O *output* do programa transformacional é uma árvore verticalizada, que pode ser usado diretamente como *input* nos programas pedagógicos gráficos Java que utilizamos no projeto VISL.

<ATENÇÃO MARCELO: falta o (7)>

<ATENÇÃO PAGEMAKER: cores>

(8a) Análise CG intratextual

<ATENÇÃO MARCELO: não seria a observação de comando – entre colchetes, portanto?>

Entre **@ADVL>** os **@>N** donos **@P<** de **@N<** restaurante **@P<,** a **@>N** crise **@SUBJ>** valoriza **@FMV** quem **@SUBJ> @#FS-<ACC** pilota **@FMV** a **@>N** própria **@>N** cozinha **@<ACC**.

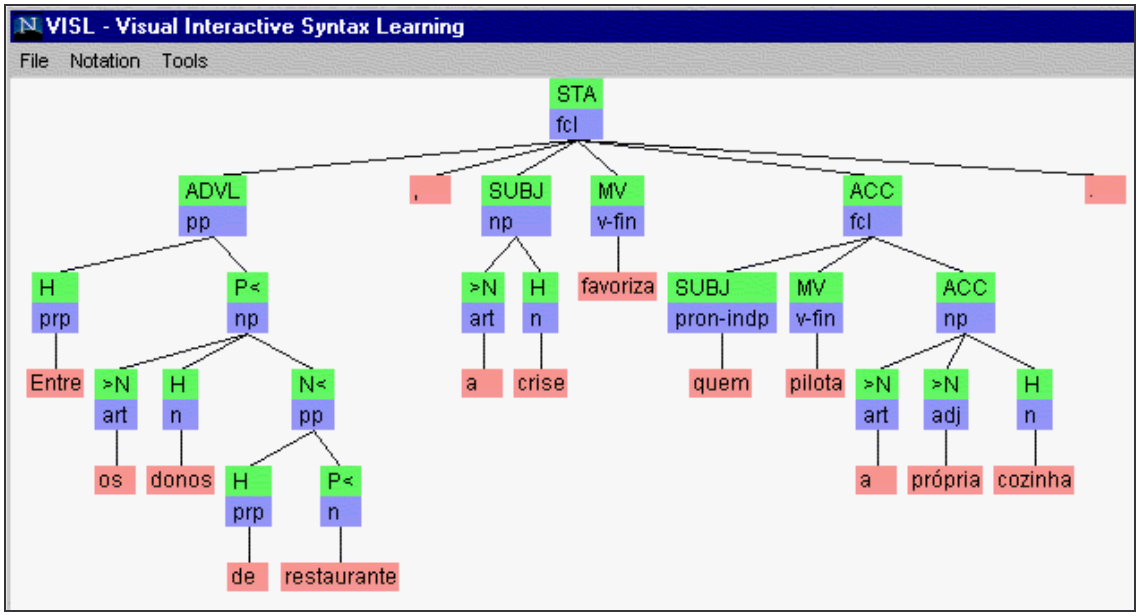
(8b) Análise transformada em árvore sintática vertical, com etiquetas novas para sintagmas

@ADVL>:pp		
|-@H:prp	Entre	[entre]

|-@P<:np		
|-@>N:DET M P	os	[o] <art>
|-@H:N M P	donos	[dono] <H>
|-@N<:pp		
|-@H:PRP	de	[de]
|-@P<:N M S	restaurante	[restaurante] <inst>
@SUBJ>:np		
|-@>N:DET F S	a	[a] <art>
|-@H:N F S	crise	[crise] <sit>
@FMV:V PR 3S IND VFIN	valoriza	[valorizar] <vt> <sN>
@<ACC:fcl		
|-@SUBJ>:SPEC M/F S/P	quem	[quem] <rel>
|-@FMV:V PR 3S IND VFIN	pilota	[pilotar] <vt> <vH>
|-@<ACC:np		
|-@>N:DET F S	a	[a] <art>
|-@>N:ADJ F S	própria	[próprio] <jn>
|-@H:N F S	cozinha	[cozinha] <ejo>

(8c) Análise transformada em árvore sintática gráfica de constituintes

<ATENÇÃO MARCELO: faltam dados do quadro, Cf. cópia em papel>



3 Avaliação quantitativa

As regras gramaticais CG do *parser* foram desenvolvidas e corrigidas usando textos muito variados: literários, científicos, jornalísticos e até transcritos de língua falada (dos

projetos NURC, Brasil, e CORDIAL-SIN, Portugal). A linguagem padrão foi originalmente Português brasileiro, mas durante o projeto o sistema foi estendido também para cobrir Português europeu.

Para quantificar a robustez e a qualidade do *parser*, fiz uma avaliação de textos anotados automaticamente pelo *parser*. A maioria desses textos eram artigos da revista VEJA, mas usei também um trecho literário ("O tesouro" de Eça de Queiroz). Os resultados foram relativamente estáveis e razoavelmente independentes do cunho e da temática do texto[2]. Os exemplos na estatística abaixo são bastante típicos[3], com as porcentagens de *correção* (definida como *recall* no caso de desambigüização total) alcançando mais de 99% em termos de classe de palavra e flexão, e 97-98% para sintaxe. Essas taxas comparam-se favoravelmente com as atualmente obtidas para inglês, por Pasi Tapainen e Atro Voutilainen (http://www.conexor.fi, 14.3.99), que referem-se a uma taxa de sucesso morfossintático (porcentagem de etiquetas corretas presentes no *output*) de 94.2-96.8% para ENGCG e 96.4-97% para FDG, com ambigüidades "residuais" de 11.3-13.7% e 3.2-3.3%, respectivamente. Várias gramáticas constritivas de morfologia, avaliando a desambigüização de classes de palavra, referem-se a um *recall* de 99.7%, embora com taxas de ambigüidade não-resolvidas muito diferentes: 3-7% para inglês (Voutilainen, 1992), 32% para estoniano (Müürisep, 1996) e 5% para sueco (www.sics.se/humle/projects/svens/projectPlan.html, 23.12.98).

Texto:	*O tesouro* cerca de. 2500 pal.		*VEJA 1* cerca de 4800 pal.		*VEJA 2* cerca de 3140 pal.	
tipo de erro:	erros	correto	erros	correto	erros	correto
classe de palavra	16		15		24	
lema & flexão	1		2		2	
erros morfológicos	17	**99.3 %**	17	**99.7 %**	26	**99.2 %**
função de palavra/sintagma	54		118		101	
função de oração subordinada	10		11		13	

[2] Uma exceção previsível são dados transcritos de língua falada, como evidenciou a avaliação do *parser* em conexão com o *corpus* NURC (Norma Urbana Culta, cf. Castilho et.al, 1989). Com esse *corpus* de linguagem falada o *parser* não-modificado, enquanto razoavelmente robusto em termos de classes de palavra (99% de correção), baixou para 91-92% de correção na etiquetagem sintática. Só depois da introdução de regras novas específicas (tratando, por exemplo, a desambigüização de marcadores de pausa e rupturas sintáticas) o desempenho do sistema alcançou 95-96% de correção na sintaxe.

[3] Mais avaliações, com resultados parecidos, encontram-se em (Bick, 1997-2).

erros sintáticos	64	**97.4 %**	129	**97.3 %**	114	**96.4 %**
erros sintáticos "locais" causados por erros morfológicos	- 27		- 23		- 28	
erros puramente sintáticos	37	**98.5 %**	106	**97.8 %**	86	**97.3 %**

Tabela 1: *Correção e tipologia de erros em textos previamente desconhecidos ao* parser

Para avaliar o desempenho do *parser* na geração de árvores sintáticas (tabela 4), um trecho de 5000 palavras (tirado da revista VEJA) foi analisado automaticamente por uma versão do *parser* adaptada à geração de árvores sintáticas gráficas, do tipo que se usa nos programas de ensino de gramática do projeto VISL. Nessa avaliação, *recall* e *precision* foram computados para as várias etiquetas sintáticas separadamente, aqui referindo aos nós das árvores sintáticas, não a palavras (como na CG tradicional). Ao contrário da notação de dependência chata, a notação em árvores torna visivelmente explícitas todas as ligações dependencionais. Além disso, algumas ambigüidades dependencionais (especialmente pós-nominais), que permaneciam subespecificadas ("underspecified") na Gramática Constritiva, tem que ser resolvidas na notação em árvore. No caso de ambigüidades verdadeiras (aqui definidas como ambigüidades não resolúveis dentro da janela sintática de uma frase), uma ligação de dependência foi julgada como correta quando ao menos uma das análises corretas sobreviveu na árvore dada.

<ATENÇÃO MARCELO: precisa ser amarelo e números em vermelho? Os Cadernos têm impressão em preto, e não em 4 cores>

	casos	precisão ("precision")			"recall"		
		etiqueta sintática (só)	etiqueta e ligação	ligação dependencional (só)	etiqueta sintática (só)	etiqueta e ligação	ligação dependencional (só)
@SUBJ	351	97.3	**97.3**	100	93.4	**93.4**	98.0
@ACC	368	95.7	**95.4**	99.7	97.2	**97.0**	98.6
@PIV	88	93.1	**93.1**	100	92.0	**92.0**	100
@ADV	19	84.2	**84.2**	100	84.2	**84.2**	100
@SC	113	92.2	**92.2**	100	94.7	**94.7**	99.1
@OC	17	100	**100**	100	82.4	**82.4**	88.2
@MV	596	99.3	**99.3**	100	99.7	**99.7**	100

@AUX	87	98.9	**98.9**	100	100.0	**100.0**	100
@AUX<	96	98.9	**98.9**	100	97.9	**97.9**	100
@ADVL	518	92.5	**91.9**	99.4	95.4	**94.8**	96.7
@PRED	48	87.0	**84.8**	97.7	83.3	**81.3**	87.5
@APP	20	84.2	**84.2**	94.7	80.0	**80.0**	90.0
@P<	911	99.3	**99.3**	100	98.9	**98.9**	99.0
@>A	45	92.7	**92.7**	100	84.4	**84.4**	84.4
@A< (PCP)	43	97.0	**97.0**	100	76.7	**76.7**	79.1
@A< (outro)	26	100	**100**	100	88.5	**88.5**	88.5
@KOMP<	10	100	**100**	100	90.0	**90.0**	90.0
@>N	1029	98.9	**98.9**	100	99.5	**99.5**	100
@N<	749	98.6	**97.1**	98.5	95.7	**94.3**	94.5
@SUB	71	100	**100**	100	98.6	**98.6**	100
@COM	17	94.1	**94.1**	100	94.1	**94.1**	100
@NPHR	60	75.0	**75.0**	100	100	**100**	100
@AS<	25	95.7	**95.7**	100	88.0	**88.0**	100
todas	5307	97.1	**96.8**	99.5	96.9	**96.6**	97.9

Tabela 2: *Geração automática de árvores sintáticas – desempenho*

As colunas coloridas (etiquetas isoladas) contêm dados refletindo diretamente o *output* da Gramática Constritiva, enquanto as colunas em negrito (etiqueta *e* ligação) mostram a queda em desempenho quando erros de ligação são contados até em casos de etiquetagem sintática correta. A terceira coluna (ligação só) reflete a qualidade de estrutura "pura", julgando a árvore mesma, não contando erros de etiquetagem funcional.

No desempenho do sistema como um todo, *recall* e *precision* convergem na marca de 97% para correção de etiquetagem, já conhecida das avaliações anteriores. O fato de que não piora muito (0.3%), quando se incluem os erros de ligação, parece provar que a transformação automática de análises CG em árvores sintáticas é, de fato, possível. Um *recall* de 97.9% e uma precisão de 99.5% para dependência/ligação *per se* sugere que a informação dependencional contida no *output* do sistema é de fato *mais robusta* do que a informação funcional (de etiquetagem sintática).

Vale notar que erros de ligação limitam-se quase totalmente aos pós-nominais (@N<) e adjuntos adverbiais (@ADVL), nenhum dos quais obedece ao *uniqueness principle.* Mas,

adjuntos adverbiais são problemáticos por causa de sua posição relativamente livre na frase, e modificadores pós-nominais (freqüentemente em forma de sintagmas preposicionais) notoriamente levam ambigüidade em relação à hierarquia de ligação (fato que pode ser subespecificado na notação chata da CG tradicional).

4 O parser

4.1 A estrutura hierárquica do *parser*

<ATENÇÃO MARCELO: é necessário manter as cores? Note que a figura está ligeiramente diferente do original em papel>

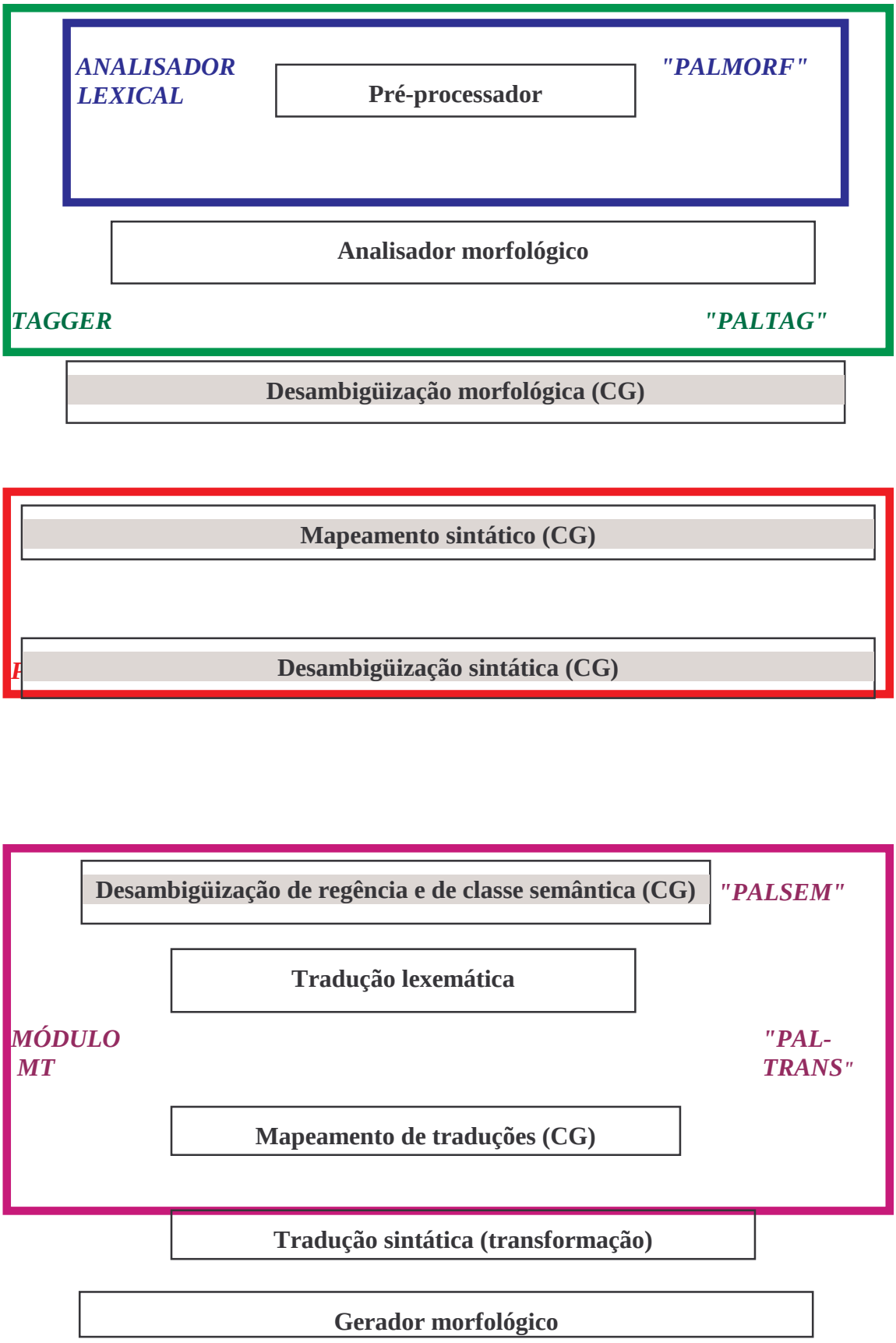


Figura 1: *Estrutura modular hierárquico do* parser *português, num sistema MT*

O *parser* português é constituído por programas modulares, que – com a exceção de um compilador de regras CG, escrito por Pasi Tapanainen (1996) – foram escritos em C e Perl especificamente para se integrarem num *parser* hierárquico. O exemplo mostra um conjunto de módulos que encaixa o *parser* em um sistema de tradução automática. Como se vê, Gramáticas Constritivas, ou de mapeamento ou de desambigüização, entram em todos os níveis intermediários, enquanto a base do sistema permanece léxico-morfológica. O nível mais alto depende do alvo aplicativo, – aqui juntam-se módulos para a resolução de polissemia[4], mapeamento contextual de traduções, transformação sintática e geração morfológica para dinamarquês. Para outras aplicações juntam-se outros módulos, como o gerador de árvores sintáticas e os programas de apresentação Java para ensino de gramática pela Internet (cap. 5).

4.2 O conjunto de etiquetas ("tag set")[5]

O analisador morfológico reconhece 13 categorias de classe de palavra, que se combinam com 24 etiquetas flexionais, resultando em centenas de etiquetas distintas e complexas. Na cadeia de etiquetagem 'V PR 3S IND VFIN', por exemplo, 'PR' (presente) alterna com 5 outras categorias de tempo, tanto no indicativo (IND) como no subjuntivo (SUBJ), e com as 6 combinações de pessoa-número existem – em sumo – 6x6x2=72 etiquetagens por verbos finitos. Portanto, só são necessárias 6+6+2=14 etiquetas individuais para descrever essa variação. Esse caráter analítico da etiquetagem torna a anotação mais compreensível e mais acessível às regras de desambigüização. Contrário a muitos outros sistemas de etiquetagem (p.ex. o sistema inglês CLAWS, descrito em Leech, Garside e Bryant, 1994), a etiquetagem discrimina estritamente entre lema, classe de palavra e categorias de flexão. Uma separação clara é mantida entre categorias morfológicas e sintáticas. Assim, as classes de palavras se definem quase exclusivamente por traços morfológicos, um substantivo (N), por exemplo, sendo uma palavra que tem *genus* como categoria lexemática (invariável) e *numerus* como categoria flexional (variável), enquanto num adjetivo (ADJ), tanto *genus* como *numerus* são categorias flexionais. Finalmente, num nome próprio (PROP), ambas são categorias lexemáticas.

O conjunto de etiquetas sintáticas contém umas 40 categorias. Aproximadamente 10 delas se usam não só para palavras e sintagmas, mas também para orações subordinadas.

---

[4] Esse módulo do *parser* utiliza desambigüização CG de traços e protótipos semânticos, como também o mapeamento de significados através de etiquetas morfossintáticas desambigüizadas. Parte do sistema foi descrito em (Bick, 1997-2).

[5] Uma lista completa de etiquetas morfológicas e sintáticas encontra-se em http://visl.hum.sdu.dk.

Nesse caso, a etiqueta contém mais um marcador de *forma sintática oracional* (oração finita, infinita ou averbal), criando um conjunto de 30 etiquetas diferentes (p.ex., @#ICL-SUBJ para uma oração infinita com a função de sujeito).

No nível semântico, existem etiquetas para aproximadamente 200 protótipos semânticos para substantivos, cobrando diferentes combinações de 16 traços semânticos atômicos. No caso de adjetivos e verbos, o conjunto de etiquetas semânticas atual é mais pobre, reconhecendo só a distinção de ±HUM para o substantivo modificado (no caso de um adjetivo) ou para o sujeito (no caso de um verbo).

Finalmente, o *parser* usa mais de 100 marcadores de potencial de valência, fornecidos no léxico para ajudar a desambigüização sintática, e desambigüizados eles mesmos no nível semântico.

5 A aplicação pedagógica

No projeto VISL, o *parser* português aqui descrito serve como núcleo de um conjunto de programas (Bick, 1997-3), que tem como objetivo ensinar gramática portuguesa pela Internet (http://visl.hum.sdu.dk). Além disso, o sistema português funciona como modelo para aplicações semelhantes em outras línguas (inglês, alemão, espanhol, francês, italiano, dinamarquês etc.). O projeto VISL é baseado na Universidade de Odense e atraiu apoio econômico de várias instituições dinamarquesas de ensino e pesquisa. Na fase atual, o sistema está sendo adaptado ao uso em outros meios universitários e não-universitários, como os do ensino escolar de primeiro e segundo graus. Como se pode imaginar, o trabalho com língua portuguesa feito pelos pesquisadores do VISL – para o ensino básico – é um luxo para um país como a Dinamarca, visto que, nesse nível, as escolas dinamarquesas ensinam somente inglês, alemão e francês. Ainda assim, o apoio à pesquisa é mantido e, ultimamente, também tem nos ajudado o interesse lusófono pelo meu Projeto. Gostaria de destacar que, por intermédio do Projecto Processamento Computacional do Português, uma fundação portuguesa para ciência e tecnologia está oferecendo financiamento para três bolsistas (dois brasileiros, uma portuguesa) de pós-graduação virem trabalhar no Projeto em Odense.

O sistema VISL permite tanto a apresentação como a construção interativa de estruturas gramaticais em frases pedagogicamente escolhidas e pré-analisadas, ou, no caso de português, dinamarquês, espanhol e inglês, num texto qualquer em linguagem natural. Ao mesmo tempo o sistema permite uma certa variação e flexibilidade em termos de tradição gramatical, uso de símbolos e categorias, e nível de complexidade.

Por exemplo, a gramática portuguesa VISL permite três sistemas de classificação de pronomes em paralelo:

(a) a tripartição morfológica em pronomes flexionantes (pron-dep), não-flexionantes (pron-indp) e pessoais (pron-pers);

(b) a distinção sintática entre pronome pré-nominal intra-sintagmático (DN:pron) e pronome independente (por exemplo, S:pron);

(c) a classificação tradicional sintático-semântico em 6 categorias de pronomes e outra de artigo: pronome interrogativo (pron-int), relativo (pron-rel), pessoal ou reflexivo (pron-pers), possessivo (pron-poss), demonstrativo (pron-dem) e indefinido (pron-indf), artigo definido (art-def).

Um exemplo de exercício gramatical interativo é o jogo de cores gramaticais: etiquetas de classe de palavra, fornecidas pelo *parser*, podem ser usadas para colorir um texto, palavra por palavra, usando vermelho para verbos, azul para substantivos, verde para adjetivos, amarelo para advérbios etc. Embora criado principalmente para níveis básicos de ensino, o programa permite também a etiquetagem de função sintática (sujeito, objeto direto), a qual junta-se às palavras coloridas em forma de "subscripts" ou "superscripts"[6]:

[6] A anotação intratextual por cores e índices de função sintática também se oferece à apresentação de *corpora* anotados, permitindo uma orientação visual na estrutura gramatical do texto e ao mesmo tempo preservando a sua legibilidade como texto corrente.

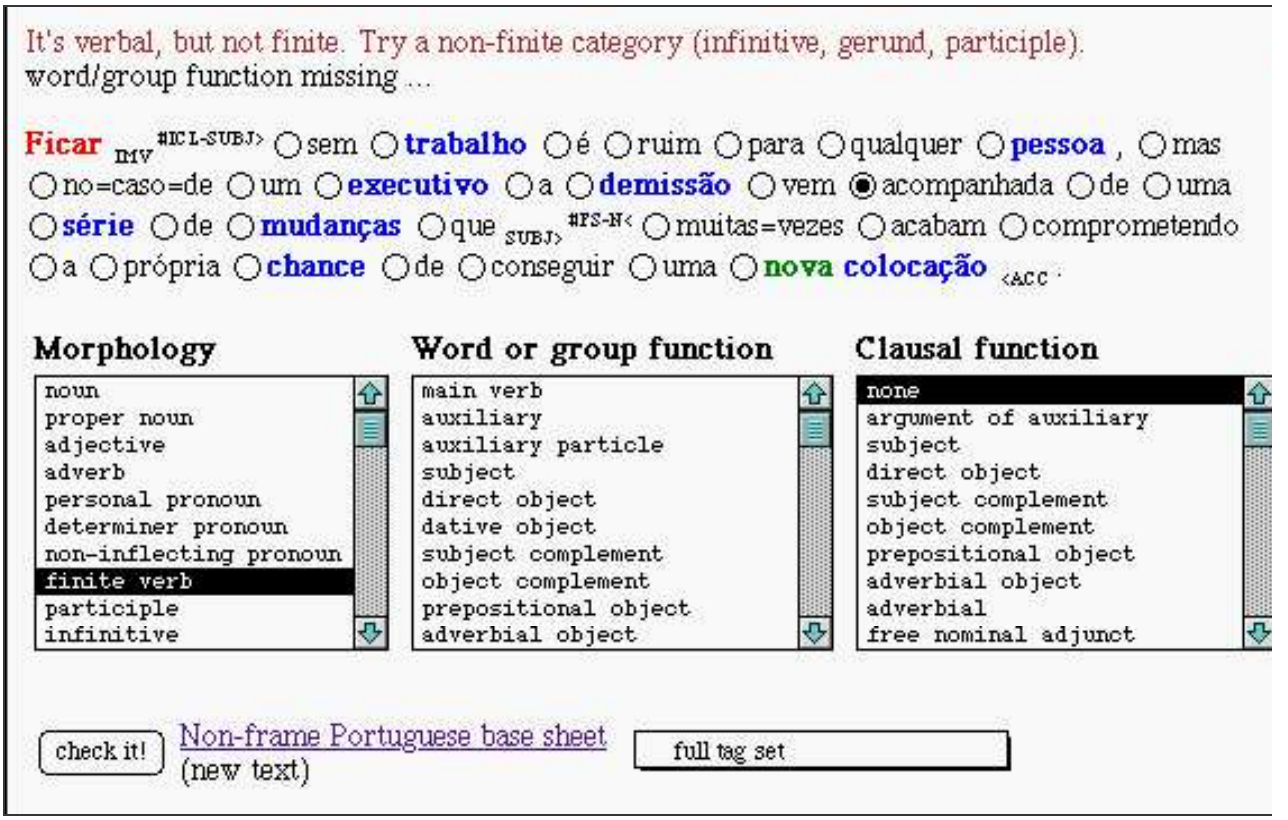


Figura 2: *Anotação intratextual, marcação e tutelagem interativa*

No exemplo, o estudante, clicando os botões e escolhendo os menus, já coloriu um número de substantivos e marcou a função (de sujeito) de uma oração subordinada infinita na sua raiz verbal, mas errou quanto à subclasse morfológica da palavra "ficar". Nesse caso, o programa tutela a escolha "semi-errada", aceita a classe de palavra (verbo) e propõe alternativas em termos de subclasse (infinitivo, gerúndio, particípio). De tal maneira, distinguem-se entre "erros absolutos" – não aceitáveis – e "erros relativos" – aceitos, mas comentados. A última escolha em cada menu é um botão "mostra-me" que revela as etiquetas certas e ajuda a colorir palavras difíceis. Foi necessário introduzir essa escolha por causa do grande número de etiquetas especialmente na sintaxe – e naturalmente também para não frustrar desnecessariamente as ambições do aluno. A opção "mostra-me" também resolve os poucos casos em que o sistema errou, e não o aluno, situação rara mas previsível quando se trabalha com linguagem natural não controlada.

Linguagem não-controlada é certamente prova da eficiência do sistema, mas nem todos alunos de uma língua estrangeira gostam de criar os seus próprios enunciados e não tem certeza que tal enunciado seria gramaticalmente correto. Até a possibilidade de copiar um texto diretamente de uma janela de *corpus*, ou da *website* de uma revista, pode tornar-se chata a longo prazo. Por isso, introduzi a opção de pedir enunciados randomizados tirados

automaticamente de um *corpus*, usando pontuação para repartição em orações e rejeitando trechos sem verbo finito. Esse módulo funciona como um jogo de auto-exame, e a idéia será em um futuro muito próximo, produzir jogos verdadeiros baseados nessa técnica, por exemplo: "*shoot the verb*" (atira no verbo) ou "*paint brush*" (pincel).

O programa de ensino atualmente mais usado é uma aplicação Java que permite inspecionar e manipular árvores sintáticas de maneira gráfica. Quando se trabalha, nessa aplicação, com texto não pré-analisado, a análise CG é aumentada por um módulo especial de identificação de limites e tipos de constituintes:

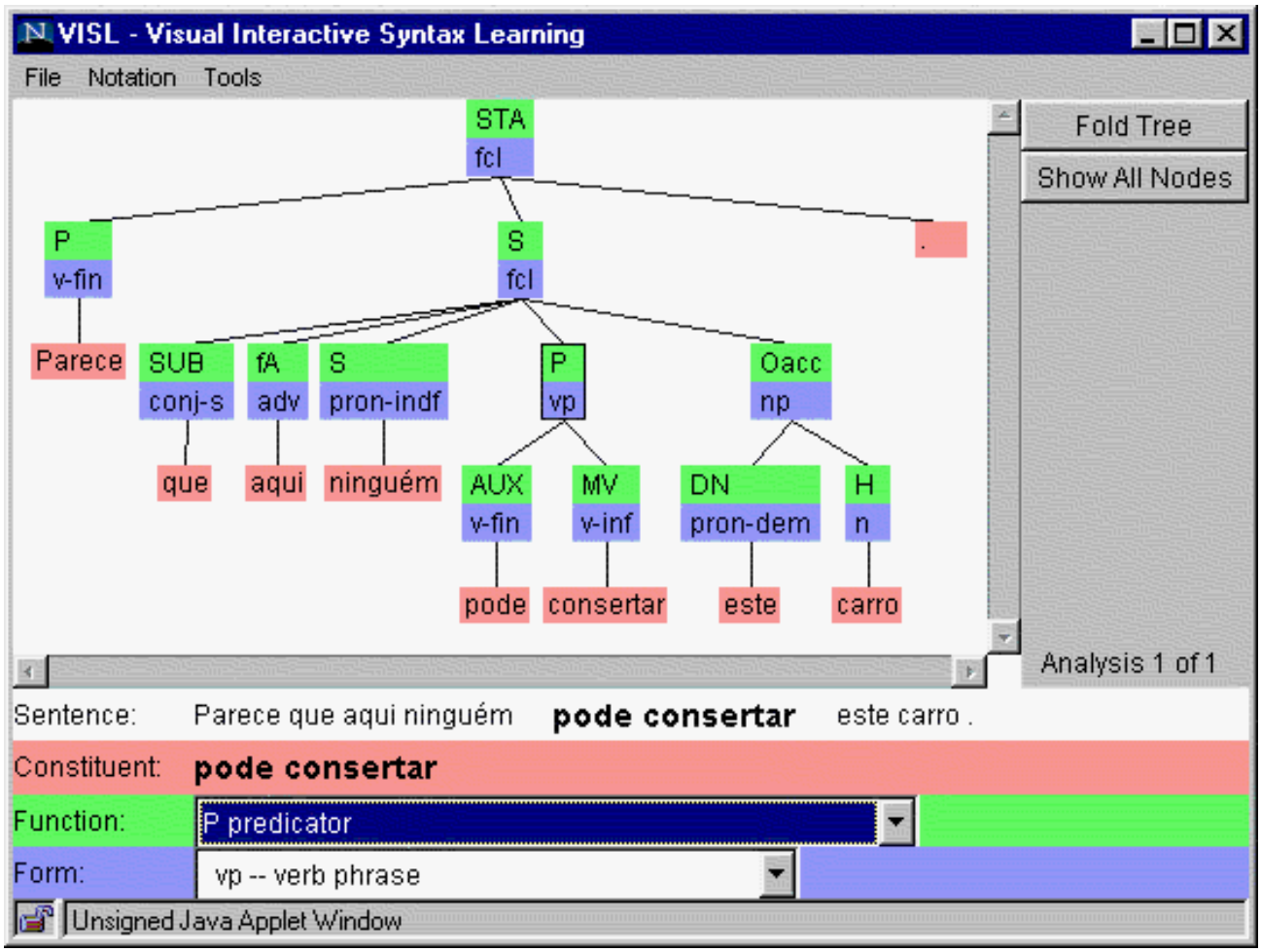


Figura 3: *Estruturas sintáticas em árvores (anotação VISL)*

Na árvore, cada nó tem uma etiqueta de *forma* (classe de palavra, tipo de sintagma ou oração), e uma de *função* sintática (p.ex. Oacc=objeto direto acusativo, fA=adverbial livre), função é marcada em cima, forma, embaixo. Nesse caso, o usuário do programa optou pelos símbolos padrões do projeto VISL, as etiquetas originais CG sendo substituídas por outras. Por exemplo, todos os dependentes no nível dos sintagmas marcam-se por D (dependente), e @>N torna-se DN (dependente adnominal). Mais importante, a estrutura mesma da árvore

mudou, porque um novo tipo de constituinte foi introduzido, – o predicador (P), que funciona como nó central para os verbos auxiliares (AUX) e principais (MV) da análise chata CG. Uma terceira opção notacional, mais simples, não reconhece predicadores, nem tipos diferentes de dependentes (D), sintagmas (g=group) e orações (cl=clause).

Para trabalhar mais sistematicamente com assuntos gramaticais bem-definidos, criamos o que chamamos de "*corpus* fechado" para cada língua VISL. Um *corpus* fechado é um conjunto de enunciados morfossintaticamente pré-analisados e "manualmente" controlados, com capítulos a encaixarem-se nas temáticas do ensino tradicional (tipo quadro e giz), e com uma certa progressão pedagógica de material simples para enunciados e estruturas mais sofisticadas. Atualmente, estamos introduzindo uma tripartição do material de *corpus* fechado em três níveis de dificuldade, (a) universidade, (b) ginásio (escola secundária) e (c) escola primária. De um certo grau, superpõe-se a essa distinção outra, a entre ensino gramatical de língua materna de um lado e língua estrangeira do outro lado. No caso do Português na Dinamarca, lamento ter de admitir que o país só oferece a língua de Camões como língua estrangeira e ao nível universitário. Por isso, nessa altura, a tripartição do material serve simplesmente para fornecer exemplos adequados tanto para estudantes novos como para estudantes já mais experientes.

Outra maneira de exemplificar assuntos gramaticais específicos, no âmbito VISL, é através de *corpora* verdadeiros (chamamos esses de "abertos"), que podem ser analisados "live" pelo *parser*, e depois submetidos a uma busca automática. Assim, podemos tirar do livro de exemplos caótico que é um *corpus* um capítulo mais coerente de material relevante. Imaginamos, por exemplo, que um estudante tenha uma dúvida em relação a cadeias verbais em Português – ele não tem certeza se podem ou não integrar-se à cadeia preposições interpostas aos verbos. Ele então escolhe "open corpus search" e formula uma busca de preposições depois de auxiliares (@FAUX_PRP) ou antes de complementos auxiliares (PRP_@#ICL-AUX<). O sistema agora vai criar um capítulo volumoso sobre o assunto, e o nosso estudante, se calhar, vai concluir que a distribuição desse fenômeno gramatical não permite orações subordinadas com função de sujeito e resolve buscar um contra-exemplo específico: @#ICL-SUBJ>_PRP_@#ICL-AUX<

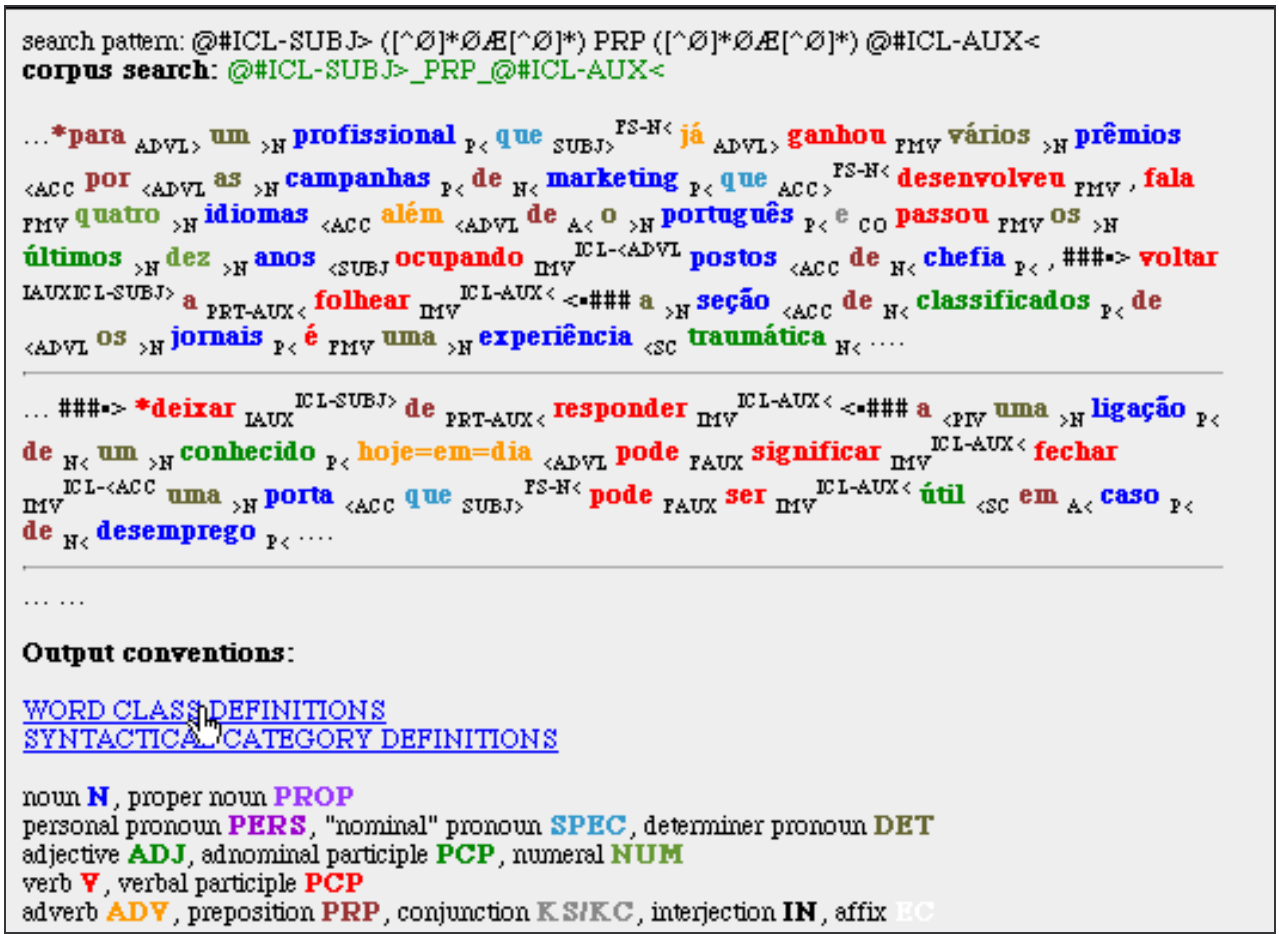


Figura 4: *Busca em* corpora *anotados*

Aqui, foram encontrados dois exemplos numa busca bastante específica. O sistema marca a estrutura alvo por flechas (###> <###), e a anotação tipo "texto enriquecido", usando cores e *subscripts* para forma e função, permite ao mesmo tempo coerência textual e visibilidade de categorias e estruturas gramaticais. Até etiquetas não visíveis no *output*, por exemplo 'plural' (P), 'transitividade' (<vt>) ou a base lexemática podem ser buscadas: qualquer combinação de palavra, forma de base, categoria, flexão e função sintática é aceita como busca legítima.

Também se pode buscar, no *corpus* fechado, o *output* sendo apresentado em árvores sintáticas com nós de forma e função e, naturalmente, os exemplos encontrados aqui podem ser manipuladas gráfica e interativamente. Para experimentar, juntei também o material do projeto CORDIAL-SIN, de língua portuguesa dialetal falada, que foi analisado pelo *parser* e transformado em árvores sintáticas. Para o ano que vem planejo a criação de um verdadeiro "banco de árvores" (*tree bank*) para Português, com milhares de palavras, e uma interface semelhante a do sistema VISL.

6 Conclusão e perspectiva

"*Parseres"* baseados em Gramática Constritiva são robustos e podem alcançar porcentagens de erro muito baixas na análise morfossintática de textos livres de linguagem natural. O método implementa-se com uma notação descritivamente elegante, com etiquetagem de palavras, que dentro do mesmo formalismo pode tratar diferentes níveis de análise, tanto de morfologia como de sintaxe e semântica. Para Português foi possível implementar também a análise automática de orações subordinadas, permitindo, junto com um sistema elaborado de marcadores de dependência, a transformação da notação chata (por etiquetagem) em árvores sintáticas (de constituintes).

A incorporação do *parser* no sistema de ensino gramatical VISL mostra como um *corpus* fechado de exemplos pedagógicos pode tornar-se num *corpus* aberto de linguagem natural anotada "*on the fly*", e como uma gramática de análise automática permite filtragem notacional e modificação estrutural para adaptar-se a outros paradigmas descritivos. Na Dinamarca, a parte portuguesa do sistema VISL foi introduzida no ensino de gramática portuguesa no meio universitário há três anos e tanto o sistema gramatical (CG) como o conjunto de programas foram adaptados para várias outras línguas.

Bibliografia

Bick, Eckhard (1996). *Automatic Parsing of Portuguese.* In García, Laura Sánchez (ed.), *Anais / II Encontro para o Processamento Computacional de Português Escrito e Falado.* Curitiba: CEFET-PR.

Bick, Eckhard (1997-1). *Dependensstrukturer i Constraint Grammar syntaks for portugisisk.* In Brøndsted, Tom & Lytje, Inger (eds), *Sprog og Multimedier.* Aalborg: Aalborg Aalborg University Press

Bick, Eckhard (1997-2) *Internet Based Grammar Teaching*, in: Christoffersen, Ellen & Music, Bradley (eds.), *Datalingvistisk Forenings Årsmøde 1997 i Kolding, Proceedings,* pp. 86-106. Kolding: Institut for Erhvervssprog og Sproglig Informatik, Handelshøjskole Syd

Bick, Eckhard (1998), *Structural Lexical Heuristics in the Automatic Analysis of Portuguese*, in: *The 11th Nordic Conference on Computational Linguistics (Nodalida '98), Proceedings.* Copenhaguen: Center for Sprogteknologi (CST) and Department of General and Applied Linguistics (IAAS), University of Copenhaguen

Bick, Eckhard (1999), *Tagging Speech Data – Constraint Grammar Analysis of Spoken Portuguese*, in Lindberg, Carl-Erik & Lund, Steffen Nordahl (eds.), *17th Scandinavian*

Conference of Linguistics, Proceedings, vol. I, pp.12-30, Southern Denmark University Odense

Bick, Eckhard (2000-1), *Portuguese Syntax (Teaching Manual)*, http://www.portugues.mct.pt/Repositorio/Bick_Portuguese_Syntax3.doc

Bick, Eckhard (2000-2), *The Parsing System "Palavras" – Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Famework*, Aarhus: Aarhus University Press

Castilho, Ataliba Teixeira de (ed.) (1989). *Português culto falado no Brasil.* Campinas: Editora da Unicamp

Karlsson, Fred (1990). *Constraint Grammar as a Framework for Parsing Running Text.* In Karlgren, Hans (ed.), *COLING-90: Papers presented to the 13th International Conference on Computational Linguistics,* Vol. 3, pp. 168-173. Helsinki: RUCL

Karlsson, Fred, et. al. (1995). *Constraint Grammar, A Language-Independent System for Parsing Unrestricted Text.* Berlin: Mouton de Gruyter.

Leech, Geoffrey & Garside, Roger & Bryant, Michael (1994). *The Large-Scale Grammatical Tagging of Text.* In Oostdijk, Nellike & de Haan, Pieter (ed.): *Corpus-Based Research into Language.* pp. 47-64. Amsterdam.

Müürisep, Kaili. (1996) *Eesti keele kitsenduste grammatika süntaksianalüsaator* (Syntactic parser of Estonian Constraint Grammar), Master thesis. Tartu: University of Tartu, Institute of Computer Science

Santos, Diana & Eckhard Bick (2000). *Providing Internet access to Portuguese corpora: the AC/DC project*, in Maria Gavrilidou et al. (eds.), *Proceedings of the Second International Conference on Language Resources and Evaluation*, LREC 2000 (Athens, 31 May-2 June 2000), pp.205-210.

Tapanainen, Pasi (1996). *The Constraint Grammar Parser CG-2.* Publication No. 27. Helsinki: Department of General Linguistics, University of Helsinki

Voutilainen, Atro & Heikkilä, Juka & Anttila, Arto (1992). *Constraint Grammar of English, A Performance-Oriented Introduction,* Publication No. 21. Helsinki: Department of General Linguistics, University of Helsinki

Voutilainen, Atro (1994). *Designing a Parsing Grammar.* Publications No. 22. Helsinki: Department of General Linguistics, Helsinki University