

Gramática Constritiva na Análise Automática de Sintaxe Portuguesa

Eckhard Bick

Institute of Language and Communication, SDU - Odense University, Denmark

lineb@hum.aau.dk

<http://visl.hum.ou.dk>

Key words:

Natural Language Parsing (NLP), Constraint Grammar (CG), Visual Interactive Syntax Learning (VISL)

Resumo

Neste trabalho apresentam-se um analisador automático (tagger-parser) para português escrito ou transcrito, e algumas aplicações nele baseadas, (a) uma ferramenta para tratamento de corpora e (b) uma ajuda para ensino de sintaxe através da internet. O sistema apoia-se num léxico com 50.000 lemas e milhares de regras gramaticais para fornecer uma análise completa, tão morfológica como sintática, de um texto qualquer.

Para otimizar robusteza, eficiência e recall, as ferramentas gramaticais internas do sistema foram baseadas no formalismo Constraint Grammar (Gramática Constritiva, CG), introduzido por Fred Karlsson (1990, 1995). O núcleo original de programas foi construído em cima dum parser multi-nível desenvolvido no contexto dum projeto de doutoramento na universidade de Århus (Bick 1996, 1997). Embora usando um conjunto de etiquetas (tag set) bastante amplo, o parser alcança – com textos desconhecidos - um nível de correteza (correctness) de 99% em termos de morfologia (classe de palavras e flexão), e 97-98% em termos de sintaxe.

Entre outras, dependência, estrutura de argumentos e orações subordinadas são tratadas de uma maneira inovativa, permitindo transformação automática da notação sintática chata de dependência de CG em árvores sintáticas de constituintes, mais atrativas visual e pedagogicamente. O parser utiliza informação lexical sobre regência e classe semântica, e estudos preliminares sugerem a possibilidade de desambiguação nesse nível também.

Na interface de ensino gramatical, usuários podem escolher entre vários filtros notacionais, apoiando diferentes paradigmas descritivas da língua. Exemplos são exercícios nos quais coram-se palavras para marcar sua classe, ou árvores de sintaxe gráficas construídas pelo estudante e controlado pelo computador, com etiquetas de forma e função em cada nó.

Quando usado para etiquetagem de corpora "crus" (desanotados), 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 linguística, a ferramenta de corpus pode integrar-se à interface de ensino, fornecendo ao estudante exemplos de certas estruturas gramaticais.

O sistema analisa ca. 200 palavras/sec num coputador Pentium/Linux quando trabalhando com todos os níveis. Morfologia, só, alcança 2000 palavras/sec.

1. Gramática Constritiva

A maioria das palavras em textos de língua natural é – quando vista em isolamento – ambíguo quanto a classe de palavra, flexão, função sintática, conteúdo semântico etc. Só o contexto (textual, discursivo, cognitivo) determina significado e função de cada palavra. No nível de um enunciado se pode dizer que uma palavra disambigua a outra.

A Gramática Construtiva (CG), lançada por Fred Karlsson (1990, 1995) e desenvolvida na universidade de Helsink, tenta formalizar este processo cognitivo de disambiguação num conjunto de regras que restringem – através de condições contextuais - qual das interpretações possíveis para cada palavra seja escolhida ou rejeitada. Estas regras compilam-se num parser "reducional" que - nos níveis de morfologia e semântica - seleciona a etiqueta certa. No nível sintático o parser contém regras tão produtivas como restritivas, essas mapeando etiquetas ambíguas de função sintática, estas rejeitando ou selecionando etiquetas através do contexto.

Antes de embarcar no processo de disambiguação, uma CG necessita do input (1) dum analisador morfológico que, no caso do sistema português, trata de flexão e derivação, e controle lemas possíveis num léxico abrangente¹, donde também tira informação secundária, de regência e semântica, para a disambiguação contextual.

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

[ADV=advérbio, KS=conjunção 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 disambiguação para esta coorte é a seguinte:

- (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)]²

Depois da disambiguação morfológica, o parser acrescenta funções sintáticas típicas para cada palavra, usando só um mínimo de condições contextuais³:

- (3) "<nunca>"

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

² A regra foi simplificada, presumindo que cada enunciado contém ao menos 1 verbo finito – o que não sempre é o caso, p.e. em cabeçalhos, linguagem falada etc. Para tornar a regra mais segura, se pode exigir um ponto final (*1 PONTO), ou aproveitar a relação de regência entre o verbo transitivo '*comer*' e o constituinte nominal '*peixe*': (0 <vt>) (1C NP).

³ No caso de algumas palavras ou estruturas muito idiosincráticas pode fazer sentido mapear só uma etiqueta sintática, mas isto exige muitas condições contextuais, e frequentemente parece menos eficaz, porque a combinação de um mapeamento menos preciso com regras gerais de disambiguação pode resolver a função de muitas palavras da mesma classe ao mesmo tempo.

```

        "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*, através de uma regra SELECT aproveitando a transitividade do verbo (<vt>), mas mais típica na tradição CG – porque mais robusta – é a seleção *negativa*, onde a etiqueta certa simplesmente é a que sobreviva 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 3. 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 não existe um objeto direto na frase]⁴

Gramáticas constritivas no nível morfológico hoje existem para várias línguas, inglês, alemão, suéco, finlandês, entre outras. A mais antiga é a CG inglesa, que também cobre a área da sintaxe. Uma gramática constritiva, madura, tipicamente contém milhares de regras, 1-2000 para cada nível, e destaca-se por uma grande robustez. Para o nível morfológico de inglês, Voutilainen (1992) reporta menos de 0.3% de erros quando disambiguando completamente 94-97% das palavras.

2. Árvores sintáticas na notação "chata" de CG

2.1. Forma e função sintáticas

Historicamente, a CG nasceu na análise morfológica, e a maioria das sistemas se baseia em analisadores morfológicos TWOL (Koskenniemi, 1983), as regras apoiando-se em traços morfológicos e classe de palavra. Por isso, a descrição gramatical de CG exprime-se formalmente através da palavra, juntando à palavra etiquetas não só simples, lexicais, mas também complexas, estruturais. Sintaxe "chata" é uma consequência natural disso, e também o parser português utiliza representações estruturais chatas. A descrição contém informação sobre tão *função sintática* (p.e. argumentos como

⁴ As três regras todas usam condições contextuais "ilimitadas" válidas para a janela inteira da frase:

*-1 = a condição vale da primeira palavra à esquerda e até o início da frase

*1 = a condição vale da primeira palavra à direita e até o fim da frase

O formalismo conhece também, condições contextuais "limitadas": '-2', por exemplo, refere a posição -2 visto da palavra "target" (0), quer dizer para a segunda palavra à esquerda, '3' concerne a terceira palavra à direita etc. As condições "limitadas" podem – em princípio – ser copiadas por regras n-gram do tipo usado em analisadores probabilísticos (HMM), enquanto condições "ilimitadas" (usando *) são mais específicas do formalismo de Gramática Constritiva.

@SUBJ, @ACC) como *forma sintática* (hierarquia de constituintes). Esta última exprime-se através de *marcadores de dependência* (<, >), indicando a cabeça de um dado sintagma, e estabelecendo implicitamente os limites sintagmáticos.⁵

(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 ⁶
	velhos	[velho] ADJ M P	@N<

Nesta 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 tem que "saber" só as suas relações imediatas, q.d. cabeça e dependentes. Com essa informação, o móbile sempre pode ser reconstituído inequivocamente.

No exemplo, 'muito' encontra-se na secção inferior do móbile, bastante fora do nó raiz, mas "conhece" sua ligação de adjeto adverbial (@>A) à cabeça do sintagma adjetival, 'velho'. Este "aponta" - como adjeto pós-nominal (@N<) - a cabeça do sintagma nominal, 'castelos', à esquerda, que, ele mesmo, funciona como objeto direto (@<ACC) do verbo principal, 'temos', a raiz de nosso móbile de dependência.

Mas até com marcadores direcionais de dependência, um problema particular da notação "chata" é a confusão possível de argumentos intra- e extra-sentenciais, como em '*O perigo de os inimigos atacarem era imanente*', onde tão 'perigo' como 'inimigos' carregam a etiqueta de sujeito (@SUBJ>). A minha solução foi marcar a função sintática de orações subordinadas. Como orações subordinadas em português sempre tem ou um complementizador (conjunção, pronome ou advérbio relativo ou interrogativo) ou um verbo infinito, resolvi equipar estas palavras com *duas* etiquetas, uma de função "interna" (como constituinte da oração subordinada), outra de função "externa" (marca @#), representando a função da oração subordinada como tudo (constituente frásico da oração principal). Tecnicamente, os dois tipos de etiquetas sintáticas são independentes, e podem ser disambiguados separadamente pelas regras do parser.

(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

[@FMV = verbo principal finito, @#FS-<ACC = oração subordinada finita, com função de objeto direto (acusativo) de um verbo principal à esquerda, @SUB = subordinador, @>N = adjeto pronominal,

⁵ Na sua gramática de dependência, construída em cima de uma CG inglês, Järvinen e Tapainen (1997) marcam todas as ligações explicitamente, com índices de referência numérica em cabeças e dependentes.

⁶ O 'A' no ponto do marcador de dependência em '@>A' indica um adjetivo ou advérbio como cabeça típica do modificador. Um 'N' (em @>N ou @N<) indica, analogamente, um nome (substantivo) como cabeça típica.

@SUBJ> = sujeito de um verbo principal à direita, @<SC = complemento do sujeito, V = verbo, KS = conjunção subordinativa, DET = pronome determinador, N = substantivo (nome), ADJ = adjetivo, PR = presente, IND = indicativo, 3 = terceira pessoa, M = masculino, F = feminino, S = singular, P = plural, <art> = artigo, <vq> = verbo cognitivo, <vK> = verbo de ligação (copula)]

Vejam os exemplos mais complexos: *O baque foi atenuado pelo fato de sua mulher ter um emprego que garante as despesas básicas da família.* A análise embaixo mostra como as relações de dependência montam as constituintes da frase de maneira hierárquica, através de uma sequência de junção (attachment sequence), do nível mais baixo (aqui o artigo 'a') até o nível mais alto, o núcleo verbal na oração principal ('>' significa 'junta-se a', ':' significa 'constrói'):

a > família:NP > de:PP > despesas:NP > garante:FS > emprego:NP > ter:ICL > de:PP > fato:NP > por:PP > atenuado:ICL > foi:S

[NP=sintagma nominal, PP=sintagma preposicional, FS=oração subordinada finita, ICL=oração infinitiva, S=oração principal]

Na notação vertical embaixo, *forma* sintática é suplementada por *função* sintática (marcadores @). Os caixilhos marcam (do exterior para o interior) a oração principal, um complemento passivo, uma oração subordinada infinitiva (com função de argumento de preposição), e uma oração subordinada finita (com função de atributo pós-nominal).

(7)

SUBJ	o	[o] <art> DET M S @>N
	baque	[baque] <cP> N M S @SUBJ>
VP	foi	[ser] <x+PCP> V PS 3S IND VFIN @FAUX
	atenuado	[atenuar] <vt> <sN> V PCP M S @IMV @#ICL-AUX<
PP-PASS	por	[por] <sam-> <+INF> <PCP+> PRP @<PASS
P<	o	[o] <-sam> <art> DET M S @>N
	fato	[fato] <ac> <+de+INF> N M S @P<
PP-N<	de	[de] PRP @N<
SUBJ	sua	[seu] <poss 3S/P> DET F S @>N
	mulher	[mulher] <H> N F S @SUBJ>
VP & ICL-P<	ter	[ter] <vt> <sH> V INF 0/1/3S @IMV @#ICL-P<
ACC	um	[um] <quant2> <arti> DET M S @>N
	emprego	[emprego] <stil> <ac> N M S @<ACC
SUBJ & FS-N<	que	[que] <rel> SPEC M/F S/P @SUBJ> @#FS-N<
	garante	[garantir] <vt> <-v-cog> V PR 3S IND VFIN @FMV
ACC	as	[a] <art> DET F P @>N
	despesas	[despesa] <ac> N F P @<ACC
	básicas	[básico] <jn> ADJ F P @N<
PP-N<	de	[de] <sam-> PRP @N<
P<	a	[a] <-sam> <art> DET F S @>N
	família	[família] <HH> N F S @P<

[@>N=modificador prenominal, @SUBJ>=sujeito (de verbo à direita), @FAUX=verbo auxiliar finito (cabeça da cadeia verbal), @IMV=verbo principal infinito, @AUX<=argumento de verbo auxiliar, @<PASS=adjunto de agente do passivo, @P<=argumento de preposição, @<ACC (objeto direto/acusativo (de verbo à esquerda), @N<=modificador pós-nominal, @FMV=verbo principal finito]

2.2. Transformação da "dependência chata" em árvores sintáticas

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. Seria que com um conjunto de etiquetas bastante rico, e com marcadores de dependência em todas as níveis, uma análise CG pode ser transformado numa árvore sintática? Para comprovar esta hipótese, escrevi um programa transformativo que insere na notação chata marcadores de limites de constituintes (constituent boundaries)⁷, 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). 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.

(8a) Análise de "dependência chata", notação CG:

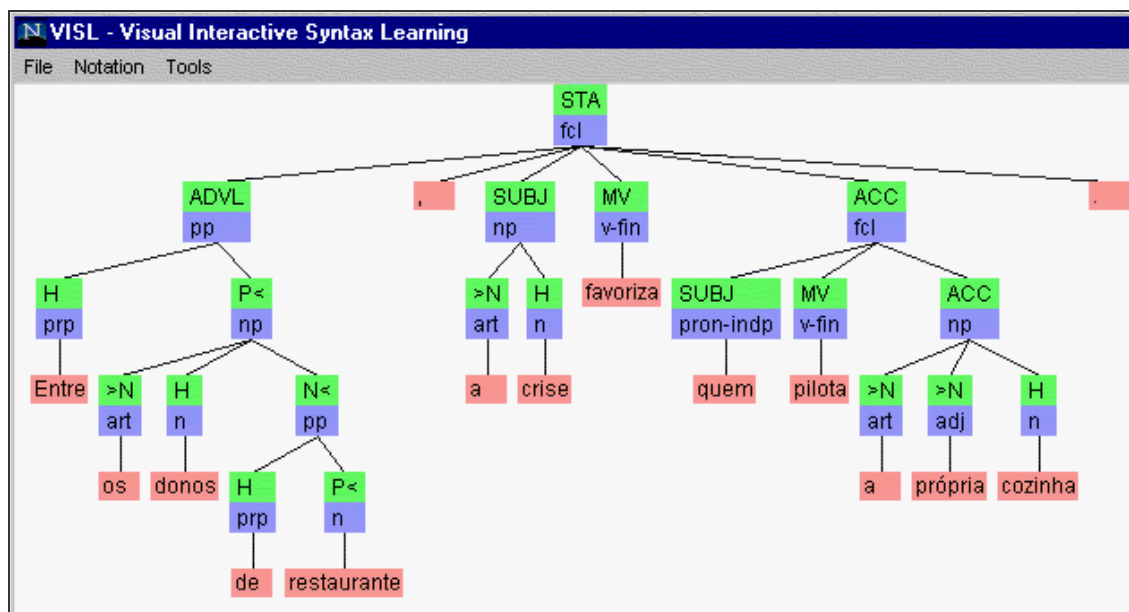
⁷ Voutilainen (1994), usa marcadores @, dentro do formalismo CG, para mapear limites de orações e outros constituintes complexos.

palavra	lema	regência e semântica	classe de palavra e flexão	sintaxe
Entre	[entre]		PRP	@ADVL>
os	[o]	<art>	DET M P	@>N
donos	[dono]	<H>	N M P	@P<
de	[de]		PRP	@N<
restaurante	[restaurante]	<inst>	N M S	@P<
,				
a	[a]	<art>	DET F S	@>N
crise	[crise]	<sit>	N F S	@SUBJ>
valoriza	[valorizar]	<vt> <sN>	V PR 3S IND VFIN	@FMV
quem	[quem]	<rel>	SPEC M/F S/P	@SUBJ>
				@#FS-<ACC
pilota	[pilotar]	<vt> <vH>	V PR 3S IND VFIN	@FMV
a	[a]	<art>	DET F S	@>N
própria	[próprio]	<jn>	ADJ F S	@>N
cozinha	[cozinha]	<ejo>	N F S	@<ACC

(8b) Análise transformada em árvore sintática vertical, com etiquetas novas para sintagmas (pp=sintagma preposicional, np=sintagma nominal, fcl=oração finita):

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

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



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 (do projeto NURC). A linguagem padrão foi originalmente português brasileiro, mas durante o projeto o sistema foi estendido para também 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 foram artigos na 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 texto⁸. Os exemplos na estatística embaixo são bastante típicos⁹, com as porcentagens de *correteza* (definida como *recall* no caso de disambiguação total) alcançando mais de 99% em termos de classe de palavra e flexão, e 97-98% para sintaxe. Estas taxas comparam-se favoravelmente com as atualmente obtidas para inglês, por Pasi Tapainen and Atro Voutilainen (<http://www.conexor.fi>, 14.3.99), que referem uma taxa de sucesso morfo-sintático (porcentagem de etiquetas corretas presentes no output) de 94.2-96.8% para ENGCG e 96.4-97% para FDG, com ambiguidades "residuais" de 11.3-13.7% e 3.2-3.3%, respectivamente. Várias gramáticas construtivas de morfologia só, avaliando a disambiguação de classes de palavra, referem unisonamente um *recall* de 99.7%, embora com taxas de ambiguidade não-resolvida muito diferentes: 3-7% para inglês (Voutilainen, 1992), 32% para estonês (Müürisep, 1996) e 5% para sueco (www.sics.se/humle/projects/svens/projectPlan.html, 23.12.98).

⁸ Uma exceção antevizível 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 este corpus de linguagem falada o parser não-modificado, enquanto razoavelmente robusto em termos de classes de palavra (99% de correteza), baixou para 91-92% de correteza na etiquetagem sintática. Só depois da introdução de regras novas específicas (trantando, por exemplo, a disambiguação de marcadores de pause e rupturas sintáticas) o desempenho do sistema alcançou 95-96% de correteza na sintaxe.

⁹ Mais avaliações, com resultados parecidos, encontram-se em (Bick, 1997-2).

Texto:	<i>O tesouro</i> ca. 2500 pal.		<i>VEJA 1</i> ca. 4800 pal.		<i>VEJA 2</i> ca. 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	
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*

Na avaliação do parser português aqui referida, modificadores (constituintes ao nível de sintagmas) foram etiquetados com marcadores de dependência (adnominais @>N, @N< e adverbais @>A, @A<), mas não entraram sub-distinções nessas categorias (como @A<PASS ou @N<PRED). Dos 54 erros sintáticos no primeiro *test run*, 13 concernem a tais modificadores, e 11 concernem a *adjuntos* no nível oracional (adverbiais @ADVL>, @<ADVL e predicativos livres @PRED>, @<PRED), enquanto quase a metade (25) são erros de função de *argumentos* no nível oracional (p.e. objeto direto @<ACC, @ACC>). Erros de função verbal (3) ou de argumentos de preposição (2) foram raros.

Intuitivamente seria lógico os erros distribuírem-se igualmente nas frases analisadas, com por exemplo menos de um erro morfológico e três erros sintáticos por cada 100 palavras. Na realidade, portanto, grande parte dos erros acontece aglomeradamente, "em grupos": palavras com erros morfológicos frequentemente figuram também na lista de erros sintáticos (especialmente erros verbo-nominais), e – além disso – "nasceram" erros sintáticos em constituintes dependentes. Subtraindo esses erros secundários de interferência morfo-sintática, a correção da análise sintática ("pura") sobe de 0.5-1%. A aglomeração de erros sublinha a importância de uma boa base lexico-morfológica em sistemas de Gramática Construtiva.

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 gramático do projeto VISL. Nesta 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). Em contrário à notação de dependência chata, a notação em árvores torna visivelmente explícitas todas as ligações dependenciais. Além disso, algumas ambiguidades dependenciais (especialmente pós-nominais), que permaneciam subespecificadas ("underspecified") na Gramática Construtiva, tem que ser resolvidas na notação em árvore. No caso de ambiguidades verdadeiras (aqui definidas como

ambiguidades não resolvíveis dentro da janela sintática de uma frase), uma ligação de dependência foi julgada de correta quando ao menos uma das análises corretas sobreviveu na árvore dada.

	casos	precisão ("precision")			"recall"		
		etiqueta sintática (só)	etiqueta e ligação	ligação dependencial (só)	etiqueta sintática (só)	etiqueta e ligação	ligação dependencial (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

Para reter a possibilidade de comparação com dados de desempenho de gramáticas constritivas mais tradicionais, e por não terem ligações de dependência "de cima", @H ("head", cabeça de sintagma) e @STA ("statement", raiz verbal da frase) não figuram na tabela em cima. De 1988 etiquetas @H e 264 etiquetas @STA, só 2 foram erradas, e incluir-las na estatística iria aumentar o desempenho desproporcionalmente.

Desta maneira, julgam-se só as etiquetas que também apareceriam numa análise CG "normal". A diferença entre o nombre de palavras, 4937, e o nombre de nós etiquetados, 5307, explique-se pelo fato de, na minha notação CG, as etiquetas de orações subordinadas serem ligadas como etiqueta número dois ou ao complementizador ou ao primeiro verbo da oração (que também retém a sua etiqueta intra-oracional, q.d. de sujeito, objeto, adverbial ou verbo principal). Embora o parser etiqueta co-ordenadores indicando os constituintes co-ordenados (p.e. <co-acc> no caso de dois objetos diretos co-ordenados), o programa de geração de árvores não utilizava esta informação na altura da avaliação, e por isso, erros em relação a ligações de parataxe não foram quantificados. Uma distinção foi feita entre predicativos *dentro de* sintagmas nominais e predicativos *livres* ao nível da oração.

As colunas coradas (etiquetas isoladas) contêm dados refletindo diretamente o output da Gramática Constritiva, enquanto as colunas em negro

(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 tudo, *recall* e *precision* convergem na marca de 97% para correteza de etiquetagem já conhecida das avaliações anteriores. O fato de que não muito piora (0.3%) quando se incluem os erros de ligação, parece prova de 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 dependencial contida no output do sistema é de fato *mais robusto* do que a informação funcional (de etiquetagem sintática).

Tem muita variação no desempenho para constituintes específicos. Por exemplo, sujeitos (@SUBJ) destacam-se por uma alta taxa de precisão e uma taxa (relativamente) baixa de *recall* (97.3% contra 93.4%), enquanto a etiquetagem de objetos diretos (@ACC) sofre de baixa precisão e alcança um *recall* alto (95.4% contra 97%), sugerindo que a gramática atual favorece @ACC e prejudica @SUBJ. Na avaliação de sua *Finite Dependency Grammar*, Pasi Tapainen e Atro Voutilainen (<http://www.conexor.fi>, 14.3.99) relatam uma relação semelhante entre sujeito e objeto direto, embora com um *recall* menor, as taxas sendo 95% contra 83% (sujeitos) e 94% contra 88% (objetos diretos).

O desempenho foi o melhor no caso da função verbal e dos subordinadores (onde a tarefa principal de disambiguação encontra-se já no módulo morfológico), e no caso dos modificadores pronominais (@>N) e argumentos de preposições (@P<). Mais problemáticos eram predicativos de objetos (@OC), aposições (@APP), predicativos livres do nível oracional (@PRED), e adjetos adverbiais pós-postos (@A<) regidos de participios atributivos (como em 'um artigo publicado, em 1992, por seu amigo'). Apostos e predicativos livres são problemáticos por não se encaixarem num padrão de regência e – também por isso – por serem os ultimos a ser rejeitados de uma regra CG, tornando-os a opção *default* em casos difíceis. Para todos os quatro etiquetas problemáticas *precisão* foi melhor do que *recall*, mas no caso de @OC a precisão foi 100% e com pós-adjetos de participios, 97%, relevando o fato que ambos (ao contrário de @PRED e @APP) são fortemente reglementados por informação léxical de regência, que através de regras CG permite a rejeição de mapeamentos errados "positivos", como argumentos, mas é inadequado para descrever a variação enorme no mundo de adjuntos livres (adverbiais e predicativos livres).

Vale notar que erros só de ligação limitam-se quase totalmente aos pós-nominais (@N<) e adjuntos adverbiais (@ADVL), nenhum dos quais obedece ao *uniqueness principle*. Mais, adjuntos adverbiais são problemáticos a causa de sua posição relativamente livre na frase, e modificadores pós-nominais (frequentemente em forma de sintagmas preposicionais) notoriamente levam ambiguidade em relação à hierarquia de ligação (coisa que pode ser sub-especificada na notação chata da CG tradicional).

4. O parser

4.1. A estrutura hierarquica do parser

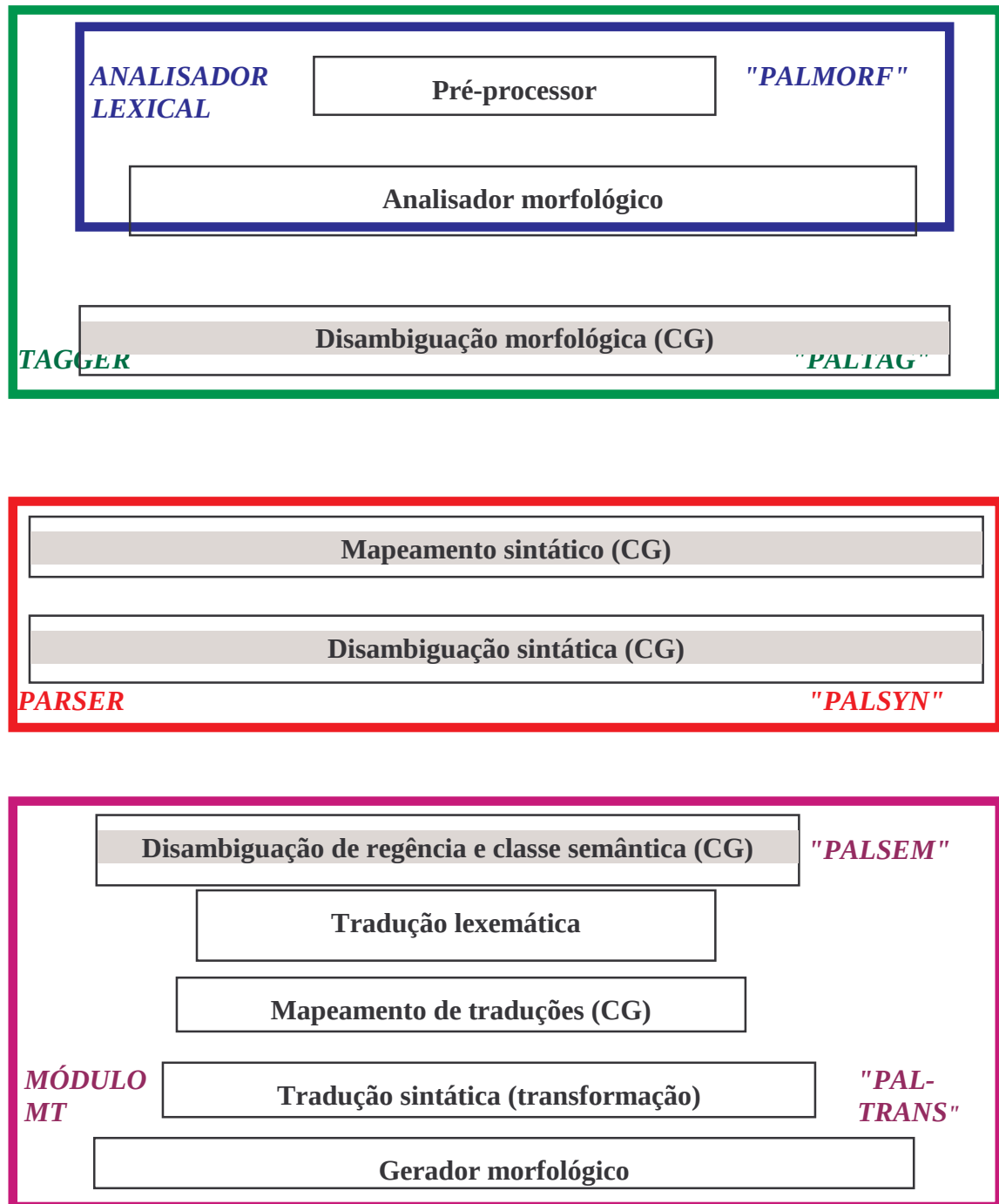


Figura 1: Estrutura modular hierarquico do parser português, num sistema MT¹⁰

O parser português constitui-se por um nombre de programas modulares, que – com a exceção dum compilador de regras CG, escrito por Pasi Tapanainen (1996) – foram

¹⁰ Machine Translation (tradução automática)

escritos em C e Perl especificamente para se integrarem num parser hierárquico. O exemplo embaixo mostra um conjunto de módulos que encaixa o parser num sistema de tradução automática. Como se vê, gramáticas constritivas, ou de mapeamento ou de disambiguação, entram em todos os níveis intermédios, enquanto a base do sistema permanece lexico-morfológico. O nível mais alto depende do alvo aplicativo, - aqui juntam-se módulos para a resolução de polysemia¹¹, mapeamento contextual de traduções, transformação sintática e geração morfológica para dinamarquês. Para outras aplicações ajuntam-se outros módulos, como o gerador de árvores sintáticas e as programas de apresentação Java de frases analisadas para ensino de gramática através da internet (cap. 5).

4.2. O conjunto de etiquetas ("tag set")¹²

O analisador morfológico reconhece 13 categorias de classe de palavra, que se combinam com 24 etiquetas flexionais, resultando em centenas de distintas etiquetas complexas. Na cadeia de etiquetagem 'V PR 3S IND VFIN', por exemplo, 'PR' (presente) alterna com 5 outras categorias de tempo, tão no indicativo (IND) como no subjuntivo (SUBJ), e com as 6 combinações de pessoa-número existem – em sumo – $6 \times 6 \times 2 = 72$ etiquetagens por verbos finitos. Portanto, só são necessárias $6 + 6 + 2 = 14$ etiquetas individuais para descrever esta variação. Este carácter analítico da etiquetagem torna a anotação mais compreensível, e mais acessível às regras de disambiguação. Contrário a muitas outras sistemas de etiquetagem (p.e. 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), tão *genus* como *numerus* são categorias flexionais. Finalmente, num nome próprio (PROP), ambas são categorias lexemáticas.

O conjunto de etiquetas sintáticas contém uns 40 categorias. Aproximadamente 10 delas se usam não só para palavras e sintágmata, mas também para orações subordinadas. Neste caso, a etiqueta contém mais um marcador, de *forma sintática oracional* (oração finita, infinita ou averbal), criando um conjunto de 30 etiquetas diferentes (p.e., @#ICL-SUBJ para uma oração infinita com a função de sujeito).

No nível semântico existem etiquetas para ca. 200 protótipos semânticos para substitantivos, 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 $\pm$ HUM para o substitantivo modificado (no caso dum adjetivo) ou para o sujeito (no caso dum verbo).

Finalmente, o parser usa mais de 100 marcadores de potencial de valência, fornecidos no léxico para ajudar à disambiguação sintática, e disambiguados eles mesmos no nível semântico.

¹¹ Este módulo do parser utiliza disambiguação CG de traços e protótipos semânticos, como também o mapeamento de significados através de etiquetas morfo-sintáticas disambiguadas. Parte do sistema foi descrito em (Bick, 1997-2).

¹² Uma lista completa de etiquetas morfológicas e sintáticas encontra-se em <http://visl.hum.ou.dk>.

5. A aplicação pedagógica

No projeto VISL (Visual Interactive Syntax Learning), o parser português aqui descrito serve como núcleo de um conjunto de programas (Bick, 1997-3), que tem como objetivo ensinar gramática portuguesa através da internet (<http://visl.hum.ou.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 exploração. 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 grão.

O sistema VISL permite tão a apresentação como a construção interativa de estruturas gramaticais em frases pedagogicamente escolhidas e pré-analisadas, ou, no caso de português, 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, etiquetas de classe de palavra, fornecidas pelo parser, podem ser usadas para corar um texto, usando vermelho para verbos, azul para substantivos, verde para adjetivos, amarelo para advérbios etc. Etiquetas de função sintática (sujeito, objeto direto) também podem ser tiradas da anotação CG, e ajuntadas às palavras coradas em forma de "subscripts" ou "superscripts"¹³:

It's verbal, but not finite. Try a non-finite category (infinitive, gerund, participle).
word/group function missing ...

Ficar _{MV} ^{#ICL-SUBJ>} ☐ sem ☐ **trabalho** ☐ é ☐ ruim ☐ para ☐ qualquer ☐ **peessoa** , ☐ mas
☐ no=caso=de ☐ um ☐ **executivo** ☐ a ☐ **demissão** ☐ vem ☒ acompanhada ☐ de ☐ uma
☐ **série** ☐ de ☐ **mudanças** ☐ que _{SUBJ>} ^{#PS-N<} ☐ muitas=vezes ☐ acabam ☐ comprometendo
☐ a ☐ própria ☐ **chance** ☐ de ☐ conseguir ☐ uma ☐ **nova colocação** _{<ACC}

Morphology	Word or group function	Clausal function
noun	main verb	none
proper noun	auxiliary	argument of auxiliary
adjective	auxiliary particle	subject
adverb	subject	direct object
personal pronoun	direct object	subject complement
determiner pronoun	dative object	object complement
non-inflecting pronoun	subject complement	prepositional object
finite verb	object complement	adverbial object
participle	prepositional object	adverbial
infinitive	adverbial object	free nominal adjunct

[Non-frame Portuguese base sheet](#)
 (new text)

Figura 2: Anotação intratextual, marcação e tutela interativa

No exemplo, o estudante, clicando nos botões e escolhendo nos menus, já corou um nome de substantivos, e marcou a função (de sujeito) de uma suboração infinita na

¹³ A anotação intratextual por cores e índices de função sintática também oferece-se 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.

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).

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

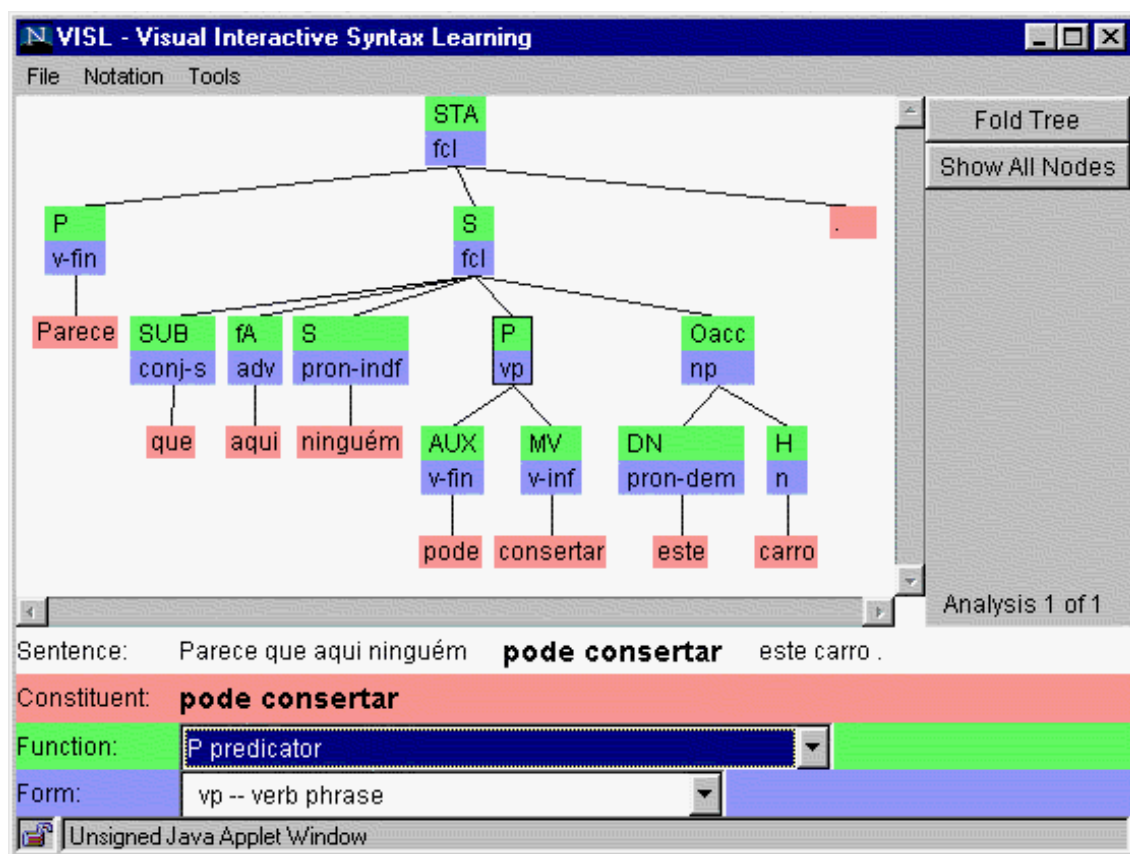


Figura 3: Estruturas sintáticas em árvores (anotação VISL)

Na árvore, cada nó tem uma etiqueta de *forma* (classe de palavra, tipo de sintagma ou oração), e uma de *função* sintática (p.e. Oacc=objeto direto acusativo, fA=adverbial livre), função em cima, forma embaixo. Neste caso, o usante do programa optou pelos símbolos padrão do projeto VISL, as etiquetas originais CG sendo substituídas por outras. Por exemplo, todos os dependentes no nível dos sintágmata 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 predicator (P), que funciona como nó mãe 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), sintágmata (g=group) e orações (cl=clause).

6. Conclusão e perspectiva

Parsers baseados em Gramática Constitutiva são robustos e podem alcançar porcentagens de erro muito baixas na análise morfo-sintática de textos livres de linguagem natural. O

método implementa-se com uma notação descriptivamente elegante, com etiquetagem de palavras, que dentro do mesmo formalismo pode tratar diferentes níveis de análise, tão 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 gramático VISL mostra como um corpus fechado de exemplos pedagógicos pode tornar-se num corpus aberto de linguagem natural anotada "on the fly". Finalmente, experimentos no nível semântico (no caso, tradução automática) indicam que o formalismo usado permite a resolução de polysemia, dum lado aproveitando informação morfo-sintática já disambiguada em níveis mais "baixos", do outro lado disambiguando protótipos semânticos e traços semânticos atômicos.

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). *Dependensstruktur i Constraint Grammar syntaks for portugisisk*. In Brøndsted, Tom & Lytje, Inger (eds), *Sprog og Multimedier*. Aalborg: Aalborg Universitetsforlag.
- Bick, Eckhard (1997-2), *Automatisk analyse af portugisisk skriftsprog*, in: Jensen, Per Anker & Jørgensen, Stig. W. & Hørning, Anette (eds.), *Danske ph.d.-projekter i datalingvistik, formel lingvistik og sprogteknologi*, pp. 22-20, Kolding: Institut for Erhvervssprog og Sproglig Informatik, Handelshøjskole Syd
- Bick, Eckhard (1997-3) *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*. Copenhagen: Center for Sprogteknologi (CST) and Department of General and Applied Linguistics (IAAS), University of Copenhagen
- Castilho, Ataliba Teixeira de (ed.) (1989). *Português culto falado no Brasil*. Campinas: Editora da Unicamp
- Järvinen, Timo & Tapanainen, Pasi (1997). *A Dependency Parser for English*. Technical Reports, No. TR-1. Helsinki: Department of General Linguistics, Helsinki University
- 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
- 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