A CG & PSG Hybrid Approach to Automatic Corpus Annotation

Eckhard Bick
Institute of Language and Communication, Southern Denmark University
lineb@hum.au.dk, http://visl.sdu.dk

Abstract

This paper describes and evaluates a hybrid non-probabilistic parsing method for the grammatical annotation of large corpora and the live analysis of teaching sentences, employing a layered scheme of lexicon- and context-based Constraint Grammars on the one hand, and Phrase Structure Grammars or syntactic bracketing algorithms on the other. The method has been fully implemented by the author for Danish and Portuguese, and to a certain degree, Spanish. Add-on-modules were also produced for existing English and French taggers. On running newspaper text, overall correctness rates (F-scores) for the two most mature systems approach 99% for word class (PoS) and 95-96% for syntactic function tags at the shallow CG-level. Though propagating CG-errors into structural errors, subsequent constituent tree analysis adds under 1% of new attachment errors on manually revised CG-input. All modules in combination, without revision, generate 50-75% structurally ”legal” syntactic trees.

1. Introduction

Large corpora of running text are beyond manual annotation or even manual revision, which is why corpus linguistics has to seek compromises between the deep analytical demands of descriptional theory and the more shallow capabilities of automatic NLP-systems, that tend to get less robust and more error prone the deeper they get. A hybrid linguistic solution to this problem is to combine a lexicon and rule driven morphosyntactic analysis with first a shallow dependency analysis and then a function based constituent analysis, drawing robustness from the former and depth from the latter.

Within the VISL project (http://visl.sdu.dk), I have designed such hybrid systems for a number of languages, in particular for Danish (Bick 2001) and Portuguese (Bick 2000), with corpus annotation being a prime applicational target. For Portuguese, over 300 million words have been annotated with CG function tags and dependency (http://acdc.linguateca.pt/accesstocorpora.html, Santos & Bick 2000), of which a 1 million word pilot chunk has undergone subsequent PSG analysis (http:/ (http://acdc.linguateca.pt/treebank/info_floresta_English.html, Afonso et.al.). For Danish, 52 million words have been annotated up to the CG-syntactic level (DSL's Korpus90/2000, Asmussen 2002 and Bick 2002, tagged searchable at http://corp.hum.sdu.dk), and a 10 million word automatic treebank is planned (http://corp.hum.sdu.dk/arboretum.html).

2. System architecture

Morphosyntactic disambiguation, function and dependency annotation are handled with Constraint Grammar tools (Karlsson et.al. 1995), which are also used to add shallow word based semantic information, like case roles or named entity types. Constraint Grammar (CG) is a methodologically based paradigm, using context based, hand written[1] rules to map and disambiguate word based tags on various levels of annotation. Most such systems, mine included, work on raw text or token/sentence separated sgml-corpora which have been preprocessed and morphologically analysed with the help of a lexicon based analyzer using inflexion, derivation and composition rules to generate so-called cohorts of possible readings (lemma, PoS, morphological features). CG-rules then discard wrong readings or select correct readings, using both close and sentence-wide context.

2.1. Morphological analysis and PoS-tagging

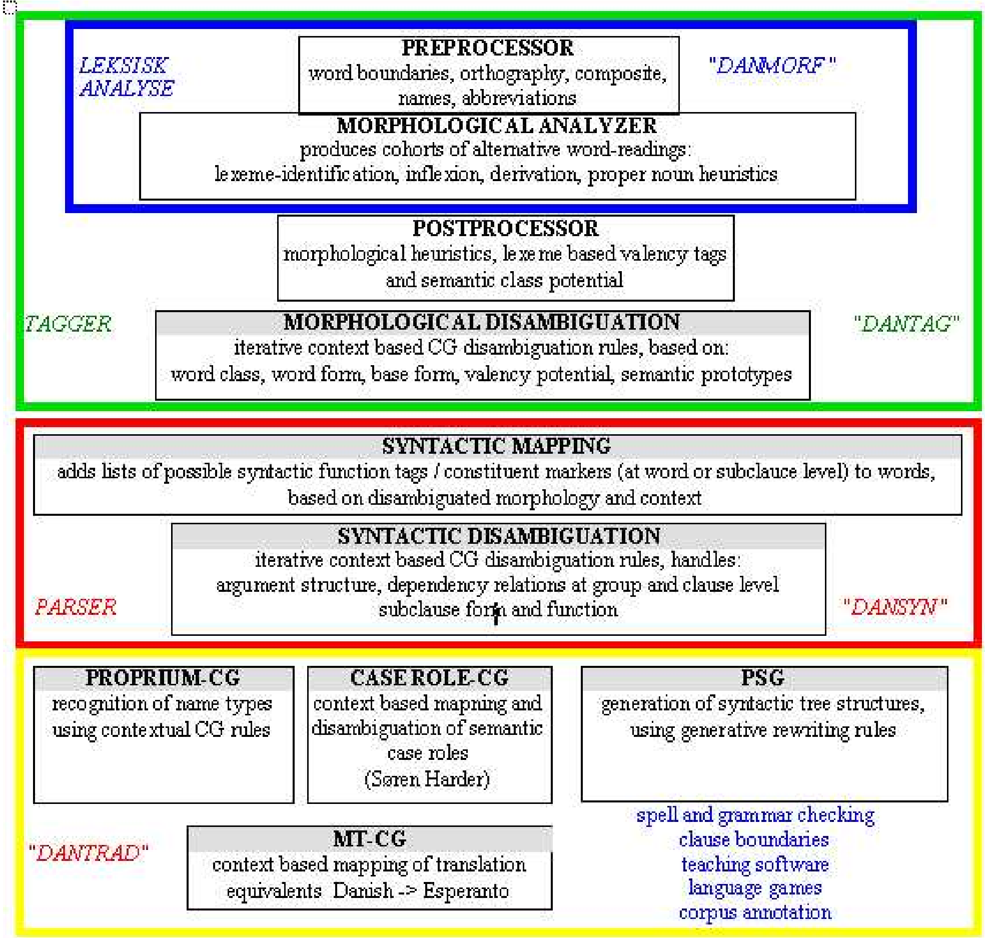
For Portuguese (1993-), Danish (1985/86 and 1998-) and Spanish (1999-), I have written morphological analyzers from scratch in C and Perl, building lexica, affix lists and preprocessors along the way. The core lexica, which were compiled from own earlier lexicographical material and corpus data, currently cover some 68.000 lexemes and polylexicals for Portuguese, 59.000 for Spanish and 101.000 for Danish, not counting irregular or other inflected forms. In the latter case, part of the lexicon was purchased from DSL (Danish Society of Literature and Language). For English, VISL has licensed Lingsoft's commercial CG system described in (Karlsson et.al. 1995), which integrates its

[1] Though all full-fledged Constraint Grammars to date have been manually written by linguists, experiments with automatically derived rules have also been performed, as in the case of the Transformation Based Learning system described in Lager (2001).

lexicon with a twol-analyzer and grammar. For French (1998-), an experimental hybrid system uses Helmut Schmid and Achim Stein's Decision Tree Tagger as input to the CG/PSG levels, mostly bypassing the lexical stage[2]. Apart from the core lexicon which contains word class and inflexion paradigm data, I use a number of specialised lexica in the mature systems, covering valency potential for the major word classes, named entity lists (20 semantic subtypes), as well as semantic prototype and atomic semantic features for nouns and adjectives and some corresponding selection restrictions for verbs. Valency and semantic information now extends to most lexemes in the Portuguese and Danish[3] lexica, while Spanish only handles some verbal valency. For English, valency potential is tagged by the Lingsoft system, while semantic prototype tags have been derived from the Princeton WordNet[4].

2.2. Constraint Grammar levels

(Figure 1: System modules)


LEKSISK ANALYSE
PREPROCESSOR
word boundaries, orthography, composite, names, abbreviations
"DANMORF"
MORPHOLOGICAL ANALYZER
produces cohorts of alternative word-readings:
lexeme-identification, inflexion, derivation, proper noun heuristics
POSTPROCESSOR
morphological heuristics, lexeme based valency tags and semantic class potential
TAGGER
MORPHOLOGICAL DISAMBIGUATION
iterative context based CG disambiguation rules, based on:
word class, word form, base form, valency potential, semantic prototypes
"DANTAG"
SYNTACTIC MAPPING
adds lists of possible syntactic function tags / constituent markers (at word or subclause level) to words, based on disambiguated morphology and context
SYNTACTIC DISAMBIGUATION
iterative context based CG disambiguation rules, handles:
argument structure, dependency relations at group and clause level
subclause form and function
PARSER
"DANSYN"
PROPRIUM-CG
recognition of name types using contextual CG rules
CASE ROLE-CG
context based mapning and disambiguation of semantic case roles
(Søren Harder)
PSG
generation of syntactic tree structures, using generative rewriting rules
"DANTRAD"
MT-CG
context based mapping of translation equivalents Danish -> Esperanto
spell and grammar checking
clause boundaries
teaching software
language games
corpus annotation

At the disambiguation and mapping stages, CG-rules can exploit contextual clues based on the one hand on such lexical information, and on the other hand on the ever-increasing syntactic and dependency information added incrementally by the grammar itself, since both are expressed in the same manner as word based tags. Furthermore, context quality increases as ambiguities are resolved and wrong potential readings discarded. As shown in the Danish example, separate CG modules are progressively applied at the morphological, syntactic and semantic levels, finally branching into specialised applicationally motivated add-ons (NER, grammar teaching, MT etc.). Also, within each CG module, rules are ordered in consecutive sections, allowing for more heuristic rules to wait for safer rules to apply first. By switching heuristic rule levels on or off, the grammar can be made to trade recall for precision. True ambiguity is simply expressed as a word retaining more than one tag of a certain

[2] A similar solution has been adopted for Italian, but since no CG module is functional yet for this language, it will not be described here.

[3] In the case of Danish, I would like to thank VISL co-workers Anders Hougaard and Lone Hegelund for their extensive contributions to the Danish valency and semantic lexica.

[4] This work was initiated by myself and continued by a VISL Ph.D.-student, Henrik Kasch.

category (say, both @N< [postnominal] and @<ADVL [adverbial adjunct], for pp-attachment ambiguity), while robustness is gained from the fact, that no rule is allowed to discard the last reading of a given kind, be it lexicon assigned or rule-mapped.

For Portuguese and Danish, my CG grammars consist of about 7000 rules for the morphological (ca. 45%) and syntactic levels (25% mapping and 30% disambiguation rules). Smaller add-on grammars handle specialised tasks, like named entity subtyping, case role assignment, spell-checker error marking or MT related tasks like polysemy resolution and translation equivalent mapping (Portuguese-Danish and Danish-Esperanto). For Spanish, the morphosyntactic core grammar was ported and adapted from Portuguese at around 4.400 rules. For English, since the original system was a ”black box” and more limited in its descriptive potential than desired, I added a CG module handling subclause function and correcting certain errors. An additional disambiguation and correction module for English was written by Ph.D.-student Henrik Kasch. The French system, though less mature than the others, is a methodological novelty in that the morphological/PoS input is based not on a lexicon/analyzer, but on an automated learning system, the intrinsic error rate of which has to be compensated for by a special intermediate ”correction CG”. Thus, this first set of rules attempts, in a context dependent way, to correct tagging errors made by the probabilistic system, or to map alternative readings for later disambiguation, while subsequent rule sets are ordinary syntactic CG modules akin to the ones described above.

3. The descriptional system: From flat dependency grammar to syntactic trees

3.1. Classical and augmented Constraint Grammar

Descriptively, classical Constraint Grammar syntax combines word based function tags with a shallow dependency description, where non-terminal function is projected onto heads. In my own grammars, I have adopted a more comprehensive system of directional dependency markers and introduced function markers for subclauses. Also, ”discontinuous” dependency is handled to a certain degree, such as interrupted quotes and raised prepositional arguments, as well as other gapping structures in Danish. Table (1) shows some of the syntactic categories used. In practice, the number is much larger, since dependency arrows (>, <, and the raising marker >>) are added and tags can combine with different types of clause markers (@FS-... finite, @ICL-... non-finite, @AS-... averbal).

Table 1: Syntactic functions

@SUBJ	subject	@CO	coordinator
@ACC	direct (accusative) object	@SUB	subordinator
@DAT	indirect (dative) object	@APP	apposition
@PIV	prepositional object	@>N	prenominal dependent
@SC	subject complement	@N<	postnominal dependent
@OC	object complement	@N<PRED	predicating postnominal
@SA	subject related argument adverbial	@>A	adverbial pre-dependent
@OA	object related argument adverbial	@A<	adverbial post-dependent
@MV	main verb	@P<	argument of preposition
@AUX	auxiliary	@>>P	raised/fronted @P<
@ADVL	adverbial adjunct	@INFM	infinitive marker
@AUX<	argument of auxiliary	@VOK	vocative
@PRED	predicative adjunct	@FOC	focus marker
@KOMP<	argument of comparative	@TOP	topic

At the syntactic level, a fully annotated Danish sentence is a verticalized chain of tokens with an ordered, unambiguous tag string each. In the example (Fig. 2), valency and semantic secondary tags have been removed.

Figure 2: CG-annotation

```
Da (When)            [da]       KS                          @SUB
den (the)            [den]      ART UTR S DEF                @>N
gamle (old)          [gammel]   ADJ nG S DEF NOM             @>N
sælger (salesman)    [sælger]   N UTR S IDF NOM             @SUBJ>
kørte (drove)        [køre]     <mv> V IMPF AKT            @FS-ADVL>
hjem (home)          [hjem]     N NEU P IDF NOM             @<ACC
```

```
i (in)             [i]       PRP                         @<ADVL
sin (his)          [sin]     <poss> <refl> DET UTR S       @>N
bil (car)          [bil]     N UTR S IDF NOM              @P<

,
kunne (could)      [se]      <aux> V IMPF AKT           @FAUX
han (he)           [han]     PERS UTR 3S NOM            @<SUBJ
se (see)           [se]      <mv> V INF AKT             @AUX<
mange (many)       [mange]   <quant> DET nG P NOM        @>N
rådyr (deer)       [rådyr]   N NEU P IDF NOM &ACI-SUBJ  @<ACC
på (in)            [på]      PRP                        @<OA
de (the)           [den]     ART nG P DEF                 @>N
våde (wet)         [våd]     ADJ nG P nD NOM              @>N
marker (fields)    [mark]    N UTR P IDF NOM             @P<
```

The sentence shows, how an np (”den gamle sælger”) is assembled around its function-carrying head (the noun ”sælger” @SUBJ>) by attaching prenominal dependents (@>N), an article (ART) and an adjective (ADJ). The subject-np arrow-points to the right, attaching to the verb ”kørte”, which again functionally represents a whole subclause as adverbial (@FS-ADVL>), ultimately right-attaching it to the auxiliary at main clause level, ”kunne”.

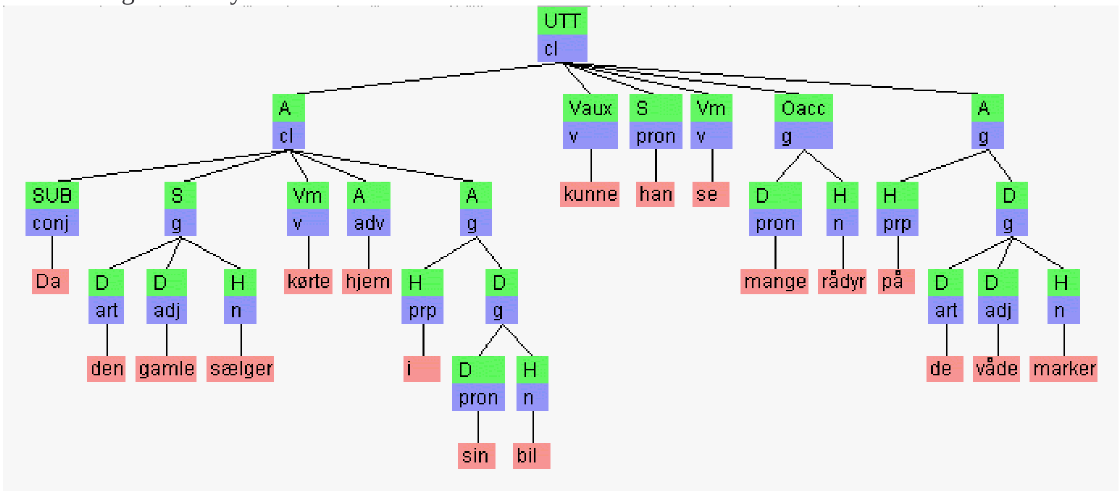
However, a flat dependency description leaves certain syntagmatic structures and dependencies underspecified, nor does it explicitly show constituent borders. Though these two types of underspecification are related, addressing them from one end or the other is an important methodological choice. Thus, one school of Constraint Grammar attacks the structural information deficit through numbered/full dependency specification, as described in Tapanainen & Järvinen (1997), where a pre-existing CG for English (ENGCG) is augmented with a Functional Dependency Grammar (FDG) module. On the other hand one could, alternatively, opt for full constituent specification as the primary method. In this vein it would make sense to combine the robustness of the CG approach with the descriptional elegance of phrase structure grammars.

3.2. Constituent tree structures on top of CG systems

Since 1996, I have therefore used various algorithms to transform CG-output into syntactic trees within the VISL grammar teaching project (Visual Interactive Syntax Learning). The oldest automatic CG-to-tree system, Portuguese, and its ”clone”, Spanish, use pattern transformation grammars compiled into a Perl program. These grammars enrich CG-output with nested opening and closing brackets for different phrase types (np, adjp, advp, finite and non-finite clauses, compound units etc.) in a context conditioned and iterative way. For Danish, English, French and - experimentally - German, I have developed a specific rule formalism for PSG-grammars, tailored to use the form and function information already contained in the CG description. In these grammars, rewriting rules use as terminals not words, but syntactic categories, dependency information, word classes and clausal categories from the CG level.

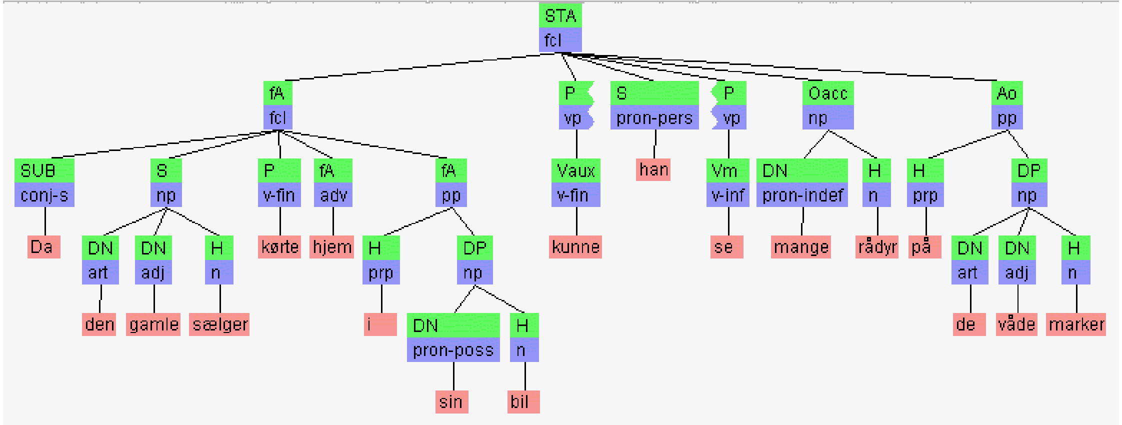
In the VISL convention, tree analyses are internally stored as vertical lists of form & function-nodes and word-terminals with indentation marking attachment depths, a format which can then be filtered into other notational variants, like the graphic format in Fig. 3a/b.

Figure 3a: Syntactic tree structure



The example tree shows both form (blue) and function (green) for each node. The category symbols used are part of a uniform VISL system employed across different languages, but can of course itself be subjected to notational filtering and category reduction depending on corpus user demands, linguistic traditions or pedagogical concerns. Thus, the simple tree (Fig. 3a) only shows two complex forms, ‘g’ (group) and ‘cl’ (clause), and uses a flat verbal chain (Vaux and Vm), while the more advanced tree (Fig. 3b) assembles a discontinuous vp-predicator, and distinguishes between np, pp and different PoS subcategories for verbs and pronouns.

Figure 3b: Syntactic tree structure



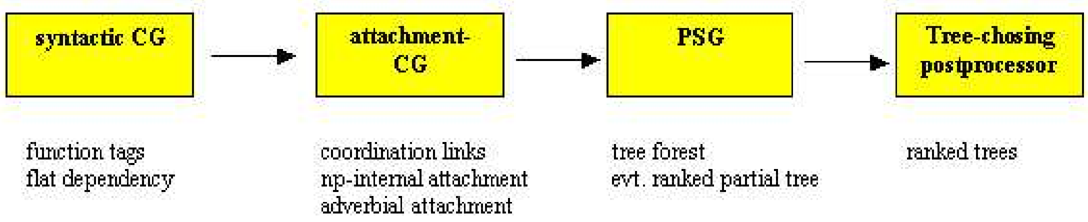
The most mature PSG module, for Danish, has about 1350 rewriting rules, while Portuguese has ca. 500 bracket mapping rules. The English and French PSG systems are smaller, with less than 200 rules each.

However, the scope of these grammars does not increase in a linear way with size. Rather, coverage shows a steep initial increase, since even a few, very simple rewriting rules of the type STA = ADVL>* SUBJ> P <DAT? <ACC? <ADVL*, together with basic np- and vp-rules handle a large percentage of sentences, given the fact that most of the work has been at the CG-level already. One factor pushing the size of the grammar is obviously the distinctional complexity of the parent CG. Thus, the Danish and Portuguese grammars have more types of adverbials, objects and postnominals than the English and French grammars, and a more fine-grained dependency description. Another factor influencing size is the way VISL’s PSG-notation handles crossing branches, since the individual parts of discontinuous constituents (like split verbal chains in germanic languages, raising, comparison dependents etc.) need their own rewriting rules. Finally there is a trade-off between space-time-consumption and grammar size: Coordination, for instance, easily overgenerates if not aided by specific function matching rules and CG-tag on coordinators as to what they coordinate. Likewise, long and close postnominal attachment in nested np’s is problematic and leads to overgeneration if not aided by additional CG-tags which again increase grammar size. While such complexity problems are quite tedious for the Danish PSG-system in long or very nested sentences, they do not affect the Portuguese pattern transformation grammar which is more ”procedural” than ”declarative”, and not fully recursive.

3.3. Trees or forests?

Another difference between the PSG solution and the pattern transformation grammar is that the latter inherently produces only one tree, while the former generates a ”forest” of possible trees, for long sentences sometimes comprising hundreds of trees. Therefore, I have written a program to rank legitimate trees, using criteria like attachment depth, coordination efficiency etc. If no legitimate tree was found (parse failure), the PSG-compiler itself ranks its partial hypotheses suggesting the tree with fewest possible ”loose ends”. Typically, parse failures can be pinpointed to where the tree - reading from left to right - breaks down to an attachment depth of zero. Of course, parse forest size will depend on the stringency of CG-input. Thus, though adverbial, object and postnominal functions of pp’s are disambiguated by the CG-grammar, it will still underspecify postnominal pp-attachment *np-internally* as a simple @N< (e.g. ”en bog om Kremls kurs i mellemøsten i anden halvdel af 1970erne” - ”a book about the Kremlin’s course in the Middle East during the second half of the seventies”). In order to restrict the number of possible trees, I have therefore in the Danish system introduced a new CG-module mapping tags for close and long postnominal attachment and adverbial attachment in layered clauses (Ill. 5). Thus ’about’, ’in’ and ’of’ should get an <np-close> tag, while ’during’ would be tagged <np-long> in the example. These and similar tags will then be used by the PSG to avoid overgeneration. The attachment module also marks coordinators for what they coordinate, since ordinary CG here allows a certain degree of underspecification in nested coordinations.

Figure 4: Forest management



5. Corpus annotation

Over the years I have used the CG annotation format (Ill. 3) in annotation tasks involving a number of different Portuguese and Danish corpora, covering both written text and transcribed speech, historical text and a wide range of genres and topics. Prominent examples are the Portuguese AC/DC-corpora (in cooperation with Diana Santos and her Linguateca team at SINTEF, Oslo) and the Danish mixed genre corpora Korpus90 and Korpus2000 (in cooperation with Jörg Asmussen, The Danish Society of Literature and Language, DSL). The word based annotation scheme allows the use of simple search algorithms and easy transformation into sgml or html mark-up, used, for instance, for colour marking PoS or named entity types. The following list illustrates the genre range of the corpora treated:

Morphosyntactically tagged

- Korpus90 and Korpus2000, mixed genre Danish text, 56M words
- DFK, mainly transcribed parliamentary discussions, Danish, 7M words
- CETEMPúblico, European Portuguese, news text, 180M words
- Folha de São Paulo, Brazilian news text, 90M words
- CORDIAL-SIN, dialectal Portuguese, 30K words
- NURC, transcribed Brazilian speech, 100K words
- Tycho Brahe corpus, historical Portuguese, 50K words

Valency tagged

- NILC corpus, Brazilian Portuguese, journalistic and essays, 39M words

Lately, inspired by the linguistic community's increasing interest in treebanks, I have used CG-to-tree systems not only to create live tree structure analyses for the VISL teaching modules[5], but also for corpus annotation, aiming at dual corpora in both CG and syntactic tree format, the Portuguese Floresta-Sintá(c)tica-corpus (launched 2001), and the Danish Arboretum-corpus (launched 2002).

Treebanks

- Floresta Sintá(c)tica, European Portuguese, 1M words (35K revised)
- Arboretum, Danish mixed genre, 70K words revised

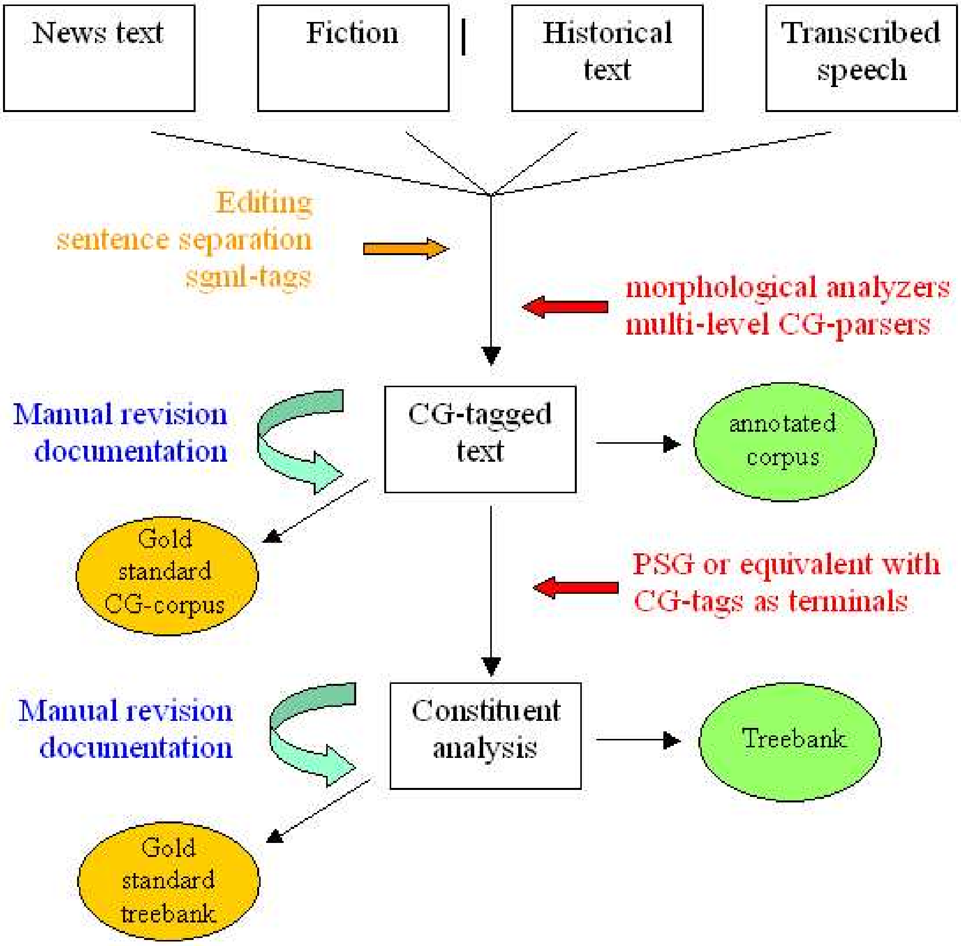
Both corpora have manually corrected parts[6], where the correction procedure matches the duality of the corpus, in that the CG-annotation is corrected before trees are generated from it, increasing the number of correct automatic trees and alleviating the subsequent correction phase at the structural/constituent level. At present, only the highest ranked tree is used in the revision process, but the quality of the ranking system should be evaluated, too, since the time consumption in revising a tree must be

[5] Small gold standard teaching corpora have been hand-coded in the same notational formalism for all VISL languages (at present 22), varying between 100 and 2000 sentences pr. language. This joint cross language annotation effort has not only provided valuable material for comparison and testing, but also helped to ensure a linguistically uniform and notationally defendable system.

[6] I would like to mention Susana Afonso and Raquel Marchi, who have been pivotal in the manual revision of the Floresta Sintá(c)tica. For their help with the manual revision of the Arboretum corpus, I would like to thank VISL student employees Ina Rasmussen and Camilla Pedersen.

balanced against the time spent on sifting through a tree forest. Figure (5) illustrates the present corpus annotation process:

Figure 5: Corpus annotation and revision procedure



6. Evaluation

6.1. Syntactic function and flat dependency

For the Portuguese (Bick 2000) and Danish parsers, I have repeatedly evaluated CG performance at the morphosyntactic level. For unknown, running, orthographically fairly "clean" newspaper text, both parsers had error rates running around 1% for PoS and 95-96% for syntactic function, depending on the granularity of the tag set[7]. A recent testrun for Danish on 195 random sentences from Korpus2000 with 2963 words, using the full tag set, produced to the following results:

Table 2

	Recall	**Precision**	**F-score**
All word classes[8]	98.6	98.7	98.65
All syntactic functions	95.4	94.6	94.9

[7] Obviously, category-fusion, like not counting dependency arrows as part of the function tag (i.e. fusing @<X and @X>), or fusing adverbial adjuncts (@ADVL), adverbial objects (@SA, @OA) and maybe prepositional object pp's (@PIV) into a common parent category adverbial (@ADVL), would increase performance in a simple mathematical way.

[8] Verbal subcategories (present PR, past IMPF, infinitive INF, present and past participle PCP1/2) and pronoun subcategories (inflecting DET, uninflecting INDP and personal PERS) were counted as different PoS.

A breakdown into different PoS categories shows that nouns, present tense verbs, prepositions, articles and personal/uninflecting pronouns are ”safe” categories (over 99% correct). Problematic for Danish are the systematic ambiguity between neuter adjective and derived adverbs (~ 96% correct), and the inflectional overlap between plural past participles (96% correct) and past tense verbs.

Table 3

Class	**recall**	**precision**	**F-score**	**Class**	**recall**	**precision**	**F-score**
N	99.5	99.1	99.2	**ART**	99.3	99.3	99.3
PROP	100	100	100	**DET**	97.1	98.5	97.7
V PR	99.2	99.2	99.2	**PERS**	99.4	99.4	99.3
V IMPF	100	97.2	98.8	**INDP**	98.2	100	99.2
V INF	98.1	99.0	98.5	**NUM**	100	100	100
V PCP1	100	100	100	**ADJ**	96.8	94.4	95.5
V PCP2	94.9	97.4	96.1	**ADV**	95.8	98.0	96.8
INFM	100	100	100	**PRP**	99.4	99.1	99.2
KS	96.6	95.0	95.7	**KC**	100	99.1	99.5

The picture is much less uniform for syntactic functions, here evaluated with a subdistinction as to attachment direction (> right, < left). Only categories with more than 5 instances in the test chunk are shown, and categories with between 5 and 10 instances have been asterisk-marked.

In the Danish CG, subjects fare better than objects, unlike the Portuguese CG (Bick 2000, p. 362), where accusative objects have a better recall than subjects (97% vs. 93.4%), a figure that is only partly compensated for by a higher precision for Portuguese subjects (97.3% vs. 95.4%). As would be expected for a language with fairly stringent word order rules, Danish right subjects (@<SUBJ) and left accusative objects (@ACC>) are both less frequent and more error prone than left subjects (@SUBJ>) and right accusative objects (@<ACC). Very safe categories are prenominal dependents (@>N) and dependents of prepositions (@P<), both ordinary and clausal (@ICL-P<, @FS-P<). Since Constraint Grammar is a reductionist method, its heuristicity can be calibrated by preserving a certain amount of ambiguity (i.e. not running the most heuristic rules). This is why some categories, most notably clausal accusative objects (@FS-ACC) and appositions (@APP and @N<PRED) have show a 100% recall, but a somewhat lower F-score.

Table 4

Class	**recall**	**precision**	**F-score**	**Class**	**recall**	**precision**	**F-score**
@SUBJ>	96.7	95.2	95.9	@>N	97.3	98.2	97.7
@<SUBJ	90.1	96.8	93.3	@N<	90.9	96.1	93.4
@F-SUBJ>	86.6	86.6	86.6	@APP*	100	87.5	93.3
@F-<SUBJ	100	100	100	@N<PRED	100	80.0	88.8
@<ACC	94.6	95.3	94.9	@>A	88.6	95.9	92.1
@ACC>*	88.8	88.8	88.8	@A<	89.4	94.4	91.8
@<DAT*	100	75.0	85.7	@P<	98.1	98.1	98.1
@<PIV	93.5	87.8	90.5	@FS-<SUBJ*	77.7	77.7	77.7
@<SC	92.0	84.3	87.9	@FS-<ACC	100	72.7	84.1
@<OC*	83.3	100	90.8	@FS-ACC>	100	91.6	95.6
@<SA	83.3	86.9	85.0	@FS-<ADVL	90.3	96.5	93.2
@<OA*	100	75.0	86.7	@FS-ADVL>	84.6	78.5	81.4
@<ADVL	93.2	90.6	91.8	@FS-P<	90.9	100	95.2
@ADVL>	96.9	93.2	95.0	@ICL-<SUBJ*	100	100	100
@KOMP<*	100	100	100	@ICL-P<	96.1	100	98.0
@P<	98.1	98.1	98.1				

6.2. Comparisons

Within the CG camp, Tapanainen & Järvinen (1997) in their work on English report function tag recall rates of 96.4% for their full dependency CG, and 94.2% for the classical ENGCG, though with a remaining ambiguity of 3.3% and 13.7%, respectively. F-scores[9] for subjects were 88.6 for news text and 94.9% for literature, F-scores for objects were 90.9 and 92.5, respectively, when evaluating not only function, but (subject/object) dependency at the same time.

Brants et. al. (1997) describe an experiment on German newspaper text (from the ECI CD-ROM), where a *probabilistic* tagger was used to assign grammatical functions to nodes in a given constituent, where both the category of the mother node and the category/PoS of the daughter were given beforehand. In this task, the tagger achieved overall 94% correct function tags (96.7% for suggested tags marked as reliable, 57.3% for tags marked as unreliable). Though language, tag granularity and text type are dividing factors, it seems worth mentioning that the Danish, English and Portuguese CG systems achieve similar or higher correctness rates *without* the benefit of manually pre-supplied PoS and structural information, possibly because hand-written CG-rules, unlike the n-gram approach, make use of global sentence context. Also, it could be significant that a sizable percentage of rules in most CG-grammars is word form or lexeme based rather than category based, and that rules exploit hundreds of lexical and mapped tags, combining in tens of thousands of unique tag strings, a complexity which might well lead to time/space- or sparse data problems in a corresponding n-gram model.

Buchholz (2002) trained Memory Based Learning (MBL) systems on converted Penn Treebank data, using PoS sequences, chunking and other information to predict grammatical relations (syntactic function). The system achieved a maximal F-score of 83.10 on that corpus (ibd. p. 220), and 73.0 on Caroll & Briscoe's training data (p. 151). Here, non-clausal direct objects performed better than subjects (93 vs. 90.5 in the first case, p. 130, and 83.2 vs. 79.6 in the second, p. 152). Like in the Danish and Portuguese CGs, the function of subclauses was addressed, and a distinction was made between direct (accusative), indirect-prepositional and indirect-dative objects.

6.3. Constituent analysis / attachment evaluation

Though it may resolve some remaining ambiguity, PSG-based tree-generation will obviously propagate errors made by the CG-levels[10]. On top of this, since it *adds* structural information, the PSG will add errors of its own, not least related to the resolution of attachment ambiguity and coordination. Since sentence length is a deciding factor, numbers vary across text types, but rough counts show that if all levels are run on free text, 50-75% of all sentences will receive a full tree analysis. These trees will typically be those where CG did not make any major word class or function tag errors, but they may still contain wrongly resolved np-internal attachment ambiguity and the like. The relative weight between propagated (CG-inherited) and PSG-internal errors can be seen when running the PSG and tree-chooser on *corrected* input rather than live CG-output. Thus, working from the hand-corrected version of the test chunk evaluated above, only 5% of sentences did not get a full tree analysis. Both complete and partial trees were evaluated for structural correctness, counting grammatically and contextually viable attachments as correct, even if they represented the subjectively less likely variant of an ambiguous structure. Under these conditions, of 4610 nodes generated for the 195 sentences, only 0.8% were wrongly attached. Of these, almost half lacked attachment to the correct np-head, a quarter were wrong clause level attachments (of otherwise correct functions) and another quarter were misattachments of conjuncts or coordinators. For comparison, the Portuguese system had a recall of 97.9% and a precision of 99.5% for dependency alone on uncorrected CG-input (Bick, 2000 p.362). Here, too, dependency error mostly concerned postnominals and adverbials.

Though evaluation methods would have to be aligned, it would be interesting to compare the presented method, i.e. attachment/constituent resolution *after* a rule-based functional analysis, with automatically learned constituent-chunking *before* a (deep or shallow) functional parse. Thus, Brants (1999) reports a maximal F-Score of 86.5 for phrase chunking with Cascaded Markov Models for raw German text, treating NP-, PP-, AP and ADVP-boundaries, but not postnominal PP-attachment or relative clauses.

7. Conclusion

I have shown, how a hybrid non-probabilistic approach can be followed for the dual annotation of large corpora, employing a layered scheme of lexicon- and context-based Constraint Grammars on the one hand, and Phrase Structure Grammars or syntactic bracketing algorithms on the other. For Danish, the

[9] Author's calculation - in the paper itself, recall and precision were given.

[10] In fact, its robusticity is much lower than a Constraint Grammar's, so often no full trees will be generated in the case of V/N errors or wrong dependency arrows on adjuncts or arguments.

two modules are interfaced by a special attachment CG and complemented by a tree ranking program. For Danish and Portuguese, on running text, correctness rates (recall at 100% disambiguation) for word class and syntactic function approach 99% and 95-96%, respectively, depending on text type and tag set granularity. At present, subsequent constituent analysis of uncorrected CG-input yields complete "legal" trees for about 50-75% of Danish sentences. On fully corrected input, 95% of sentences are assigned at least one full tree structure, with an added attachment error rate of around 0.8% after tree-ranking. Echoing the hybridity of the parsing method employed, annotated corpora come in two flavours, (a) with word based CG-tags for form, function and flat dependency (e.g. *Korpus90/2000, CETEM-Público*), or (b) as constituent tree structures with nodes marked for form and function (e.g. *Floresta Sintá(c)tica, Arboretum*). Correspondingly, manual revision is also performed in two rounds, with results from the first round of correction used as input to the PSG level. Future work should aim at minimizing the task of manual correction by adding automatic structural consistency checks. Also, since "break down points" in partial parse trees are often isolated and clearly identifiable, heuristic and incremental repair mechanisms can be imagined as a future line of research.

Afonso, Susanna & Bick, Eckhard & Haber, Renato & Santos, Diana 2002. Floresta Sintá(c)tica, a treebank for Portuguese. In González, Manuel & Suárez, Carmen Paz Araujo (eds.), *Proceedings of LREC 2002*, Gran Canaria, pp. 1698-1703

Asmussen, Jørg 2001. Korpus 2000, in *Korpuslingvistik (NyS30),* Akademisk Forlag/Copenhaguen University, Copenhaguen’

Bick, Eckhard 2000. *The Parsing System ‘Palavras’ - Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press, Aarhus.

Bick, Eckhard 2001. En Constraint Grammar Parser for Dansk. In Peter Widell & Mette Kunøe (eds.): *8. Møde om Udforskningen af Dansk Sprog, 12.-13. oktober 2000,* pp. 40-50, Aarhus University.

Bick, Eckhard 2003. Morfosyntaktisk opmærkede corpora for Dansk: Korpus90/2000 og Arboretum. In *9. Møde om Udforskningen af Dansk Sprog 10.-11. oktober 2002, Proceedings,* Aarhus University (forthcoming)

Brants, Thorsten & Skut, Wojciech & Krenn, Brigitte 1997. Tagging grammatical functions. In *Proceedings of EMNLP-2 (1997),* Providence, RI.

Brants, Thorsten 1999. Cascaded Markov Models. In *Proceedings of 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL-99),* Bergen, Norway.

Buchholz, Sabine 2002. *Memory-based grammatical relation finding.* Ph.D.-thesis. Tilburg University, 2002.

Karlsson, Fred & Voutilainen, Atro & Heikkilä, Juka & Anttila, Arto 1995, *Constraint Grammar, A Language-Independent System for Parsing Unrestricted Text,* Mouton de Gruyter, Berlin

Lager, Torbjörn 2001. Transformation-Based Learning of Rules for Constraint Grammar Tagging. Paper presented at *The 13th Nordic Conference in Computational Linguistics* (NoDaLiDa'01), Uppsala, May 21-22, 2001

Santos, Diana & Bick, Eckhard 2000. Providing Internet access to Portuguese corpora: the AC/DC project. In Gavriladou et al. (eds.), *Proceedings of the Second International Conference on Language Resources and Evaluation, LREC2000* (Athens, 31 May-2 June 2000), pp. 205-10.

Tapanainen, Pasi & Järvinen, Timo 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference on Natural Language Processing,* Washington, D.C., ACL, pp. 64-71