

Corpus-based research into language

**Edited by Nelleke Oostdijk
and Pieter de Haan**



Robust parsing of unconstrained text

Fred Karlsson

8.1 Introduction

Two of the main concerns of Jan Aarts' scholarship (e.g. 1991) have been corpus linguistics and the development of theory and methods for the automatic syntactic analysis of large, natural language corpora. A typical trait of what has come to be called the Nijmegen approach to parsing is an emphasis on the linguistic, rule-based aspects of the grammars utilized in the parsing process, in combination with careful and extensive corpus study. The TOSCA system developed in Nijmegen is a large-scale implementation of these ideas, especially in the domain of syntax (e.g. Oostdijk, 1991).

The grammatical orientation of the Nijmegen approach contrasts with recent stochastic approaches that have been successful especially in the domain of part-of-speech tagging of running English text. A correctness rate of about 96% in part-of-speech tagging is reported both for the CLAWS1 system of Garside, Leech, and Sampson (1987), and for the PARTS system of Church (1988).

In regard to automatic syntactic analysis, Oostdijk (1991: 214) reports that TOSCA assigns appropriate analyses to 88.1% (N=1517) of the 1722 sentences in a fiction sample, and to 56.17% (N=537) in a 956 sentence sample of a non-fiction text. The rest of the sentences in both samples either failed to receive an analysis altogether, or failed to receive an analysis in the set processing time limit of one hour CPU-time per sentence. Overall, TOSCA has a good syntactic coverage but it is not yet fully operational on unconstrained text.

Stochastic approaches have been less successful in the domain of syntactic analysis. CLAWS1 originally had syntactic aspirations but the goals were not fully achieved. Church's PARTS was not intended for syntactic analysis in the first place.

Robustness has been much discussed in recent parsing literature. We shall consider a syntactic parser robust if it is capable of analyzing any input, regardless of structural complexity, with reasonable success in reasonable time. Of course, the notions 'reasonable success' and 'reasonable time' are open to any number of interpretations. Optimally,

every input sentence should receive a fully correct analysis at a speed of (at least) a few thousand words per second.

Presently it is not possible to achieve any of these subgoals at the levels just stated. We shall therefore be more modest and take a system to be (fairly) robust if it satisfies the following requirements. A robust parsing system

1. does not refuse to treat any input;
2. achieves a correctness rate of some 90%, where the figure refers to the number of word-form tokens getting a unique and correct syntactic label; assessment of correctness rates presupposes testing on large amounts of running text drawn from different text types;
3. does not go wildly astray for any 'reasonable' sentence, allowing the correctness rate to deteriorate considerably;
4. achieves a minimum average working speed of 10 words per second regardless of the complexity of the input.

The purpose of this chapter is to briefly present Constraint Grammar (CG), an approach to part-of-speech tagging and surface syntactic analysis first documented in Karlsson (1990). Constraint Grammar aspires to meet the requirements (1)-(4). In particular, we shall show how the syntactic component of CG works when it is applied to English, and see what kind of syntactic output CG generates. Analyses of a set of non-trivial examples drawn from different text types will be presented. We believe that parsing theory and practice can advance significantly only when sufficient interest is paid to the treatment of everyday descriptive problems and to the enormous syntactic variation found in natural texts.

8.2 Constraint Grammar in outline

Constraint Grammar (CG) is a formalism for writing grammars that are intended to be used for parsing. The grammar statements are close to real text sentence phenomena, and they address some notorious parsing problems, especially ambiguity, in a direct fashion. The core of the CG formalism is linguistic but probabilistic elements may also be included, if the grammarian so wishes, in the form of heuristic statements working on top of the grammar-based statements. Constraints are normally formulated on the basis of extensive corpus studies.

The descriptive statements, constraints, do not have the usual task of telling what sentences are correct. The constraints are less categorical in nature, more closely tied to linguistic (especially morphological) features, and more directly geared towards the basic task of parsing, viz. bottom-up inference of morphosyntactic surface structure from a stream of words.

The constraints arrived at for language L constitute a Constraint Grammar for L. A CG for a particular language is intended to be used for parsing texts by the Constraint Grammar Parser (CGP), presently implemented in both Common Lisp (by the present author) and in C++ (by Bart Jongejan, Copenhagen and Pasi Tapanainen, Helsinki).

The input tokens to the Constraint Grammar Parser are morphologically analysed word-forms. One of the central ideas of CG is to maximize the use of morphological information. The importance of morphological features for syntactic analysis has often been underestimated.

In CG parsing, all possible structural alternatives for every word-form are assigned via lexicon, morphology, and simple mappings from morphology to syntax. The constraints discard as many alternatives as possible. The optimal result is a fully disambiguated sentence with one morphological reading and one syntactic function label remaining for each word-form.

Another central idea is to treat morphological disambiguation and syntactic labelling, i.e. the assignment of surface syntactic function labels (codes), by the same mechanism of discarding improper alternatives.

A good parsing formalism should satisfy many requirements. The constraints should be declarative rather than procedural. They should be able to cope with any real-world text-sentence, i.e. with running text, not just with linguists' laboratory sentences. They should be clearly separated from the program code by which they are executed. The formalism should be language-independent. It should be reasonably easy to implement (optimally as a finite-state automaton). It should also be efficient to run. Constraint Grammar complies with all these requirements.

In CG, the overall problem of parsing is broken up into seven modules:

- preprocessing;
- morphological analysis (including base-form reduction);
- local morphological disambiguation;
- morphosyntactic mapping;
- context-sensitive morphological disambiguation;
- determination of intrasentential clause boundaries;
- disambiguation of surface syntactic functions.

CG is the formalism of the four last stages. The same constraint formalism is used to disambiguate morphological and syntactic ambiguities, and to locate clause boundaries in a compound or complex sentence. Parts of the CG formalism are used also in morphosyntactic mapping.

Morphological analysis is relatively independent. The morphological analysers are designed according to Koskeniemä's (1983) two-level model. Currently the Research Unit for Computational Linguistics, University of Helsinki, has morphological analysers available, for English (55,000 lexicon entries), Finnish (37,000 entries), Swedish (68,000 entries), and Russian (71,000 entries).

As for details of the CG formalism that enables the grammarian to write constraints for a particular language, the reader is referred to Karlsson (1990, 1993a,b). A full English CG description has been designed by Airo Voutilainen (preprocessing module, lexicon, morphological disambiguation constraints), Heikkilä (lexicon, features), and Anttila (syntactic disambiguation constraints), cf. the papers in Karlsson *et al.* (1993), as well as Voutilainen *et al.* (1992). The full English description, all modules included, is henceforth referred to as ENGCG, short for English Constraint Grammar. All English analyses presented in this paper are due to the work of Voutilainen, Heikkilä, and Anttila.

Here are two morphologically analysed English word-forms, *a* and *move*, displayed as they look after morphological analysis, but prior to applying any morphological disambiguation or syntactic analysis:

```
("<a>"
("a" DET CENTR ART INDEF (@DN>)))

("<move>"
("move" N NOM SG)
("move" V PRES VFIN -SG3 (@+FMMAINV))
("move" V SUBJUNCTIVE VFIN (@+FMMAINV))
("move" V IMP VFIN (@+FMMAINV))
("move" V INF))
```

The word *a* has one reading, *move* five. The word-form plus the set of readings assigned to it by the morphological analyser we call a cohort. All readings in a cohort start with the base-form, i.e. base-form reduction (lemmatization) is always performed. Every reading contains morphological features, always inflectional and occasionally also derivational ones.

The designated initial character @ marks the names of all syntactic labels. The syntactic function labels in the example emanate from the lexicon (where all unique syntactic labels are introduced). For example, the label @DN> refers to a determiner occurring as modifier of some noun to the right (usually the topmost head in the current phrase). @+FMMAINV denotes a finite main verb.

A typical constraint for morphological disambiguation could state that verbal readings (i.e. readings containing the feature V) are disallowed

(i.e. discarded) if the previous cohort contains a unique instance of a determiner. This constraint would effectively disambiguate our example phrase *a move*, discarding the four verbal readings of *move* and leaving only the (appropriate) noun reading.

8.3 Syntactic parsing in Constraint Grammar

In part, this section is an abbreviated recapitulation of a part of the presentation in Karlsson (1993b), included for the purpose of facilitating interpretation of the results presented in Section 8.5.

Constraint Grammar syntax is based on dependency and assigns flat, functional, surface syntactic labels, optimally one to each word-form in the sentence (if the sentence is not truly ambiguous). Most of the labels are drawn from the classical repertoire of heads (such as subject, indirect object, object predicate complement, adverbial) and modifiers (such as pronominal adjectival modifier, postnominal modifier, prepositional complement). This is what we mean by stating that CG syntax is functional. Constituent structure plays no direct explicit role. Of course, in actual practice there is a high degree of isomorphy between dependency-oriented functional syntax of this type and constituent structure syntax.

CG syntax is surface syntax rather than deep syntax because no syntactic structure is postulated or assigned that is not in direct correspondence with word-form tokens that 'are there' in the sentence. Furthermore, the surface nature of CG syntax is emphasized by the fact that it is just an abstraction from morphological properties (drawn from lexicon and morphological analysis) and word order configurations. CG syntax maps morphological categories and word order information onto syntactic labels. In many respects, this approach relies heavily on the traditional conception of syntax, as opposed to modern conceptions that tend to emphasize the importance of constituent structure and abstract levels of representation.

The flatness of CG syntax means that no trees or other hierarchical structures are generated as a result of the parsing process. Rather, every word is assigned a syntactic code which in no obvious way differs for instance from the morphological properties of the word. Technically, the syntactic codes are just appended (as a separate list) at the end of the respective reading, as seen already in the example *move* above.

The major subclass of syntactic labels are verb chain labels (covering all verbal uses of verbs, to the exclusion of adnominal uses), (nominal) head labels, and (nominal) modifier labels. Observe, in particular, that no verb phrases are postulated. This considerably simplifies the description of complex verbal expressions as well as the description of marked word order configurations, such as the occurrence of direct objects in preverbal

position (in SVO-languages). For English, some representative examples of the three basic classes of syntactic labels are:

verb chain members:

- @+FAUXV (finite auxiliary V)
- @-FAUXV (non-finite auxiliary V)
- @+FMMAINV (finite main V)
- @-FMMAINV (non-finite main V)
- ...

nominal heads:

- @SUBJ (subject)
- @OBJ (object)
- @I-OBJ (indirect object)
- @PCOMPL-S (subject complement)
- @PCOMPL-O (object complement)
- @ADVL (adverbial)
- @APP (apposition)
- ...

nominal modifiers:

- @AN> (adjective as premodifier of N)
- @NN> (noun as premodifier of N)
- @DN> (determiner as premodifier of N)
- @GN> (word inflected in the genitive occurring as premodifier of N)
- @<NOM (postmodifier of nominal, i.e. either a noun, adjective, or pronoun)
- @AD-A> (premodifier to adjective or adverb)
- @<P (complement of preposition)
- @<NOM-FMAINV (non-finite main verb as postmodifier of a nominal)
- @<P-FMAINV (non-finite main verb as complement of a preposition)
- ...

The words in a verb chain such as *has been reading* thus get the syntactic labels @+FAUXV @-FAUXV @-FMMAINV, respectively. In the sentence *She bought the car*, the word *she* is @SUBJ, *bought* @+FMMAINV, *the* @DN>, and *car* @OBJ. These examples were presented in a form where all words had unique syntactic labels, implying that the syntactic constraints have been maximally successful in discarding all other candidate syntactic labels.

Modifier and complement labels point in the direction (right ">", left "<") of the respective head which is identified by its part-of-speech label. For instance, the label @<P is assigned to the head of a prepositional complement such as the word *park* in the expression *in the park*. The interpretation of the label @<P is that the head of the subexpression *in the park* is the next preposition to the left, i.e. *in*. The Constraint Grammar analysis of modifier and complement labels is more delicate than in traditional grammar, cf. the premodifiers @AN>, @DN>, @NN>, @GN>.

In Constraint Grammar, syntactic labels are assigned and subsequently partially discarded in three steps. The basic strategy is: Do as much as possible as early as possible. One should try to achieve a unique syntactic analysis for as many words as possible with as little overall processing as possible.

The first step is to provide syntactic labels for certain words already in the lexicon (including morphology). For word-forms having a reduced set of syntactic labels compared to what that morphological class normally has, the reduced set of labels will be listed in the lexicon. (In the English CG syntactic module, actually only unique syntactic labels are introduced in the lexicon). Thus, output from lexicon and morphology (prior to any real syntactic processing) will indicate that the word *he* is @SUBJ, that *him* is either @OBJ, @I-OBJ, or @<P (these three labels constitute a considerably reduced subset of all nominal head functions of which there are at least ten), that *went* is +FMMAINV, etc.

The second step is morphosyntactic mapping which starts syntactic processing proper. For all readings that do not have any syntactic function label due to lexicon or morphological analysis, simple mapping statements tell, for each relevant morphological feature, or combination of features, what its range of syntactic labels is. This may be compared to traditional grammar book statements such as "the syntactic functions of nouns are subject, object, indirect object, subject complement, object complement, apposition, prepositional complement, ...".

Now consider the analysis of the following sentence (borrowed from Hindle, 1989). Every word is at least two-ways ambiguous on the morphological level. Here is the analysis of the sentence after morphological analysis but prior to morphosyntactic mapping and syntactic analysis proper (output slightly simplified):

- (*her
- (she * NonMod PRON PERS FEM GEN SG3)
- (she * NonMod PRON PERS FEM ACC SG3))
- (hand
- (hand N NOM SG)
- (hand SVO SVOO INF: V SUBJUNCTIVE VFIN)

- (hand SVO SVOO INF: V IMP VFIN)
 (hand SVO SVOO INF: V INF)
 (hand SVO SVOO INF: V PRES -SG3 VFIN))
- (had
 (have INF: SVO SVOC/A V PAST VFIN)
 (have INF: SVO SVOC/A PCP2))
- (come
 (come INF: SVC/A SV P/for PCP2)
 (come INF: SVC/A SV P/for V SUBJUNCTIVE VFIN)
 (come INF: SVC/A SV P/for V IMP VFIN)
 (come INF: SVC/A SV P/for V INF)
 (come INF: SVC/A SV P/for V PRES -SG3 VFIN))
- (to
 (to PREP)
 (to INFMARK))
- (rest
 (rest N NOM SG)
 (rest SVC/A SV SVO INF: V SUBJUNCTIVE VFIN)
 (rest SVC/A SV SVO INF: V IMP VFIN)
 (rest SVC/A SV SVO INF: V INF)
 (rest SVC/A SV SVO INF: V PRES -SG3 VFIN))
- (on
 (on PREP)
 (on ADV))
- (that
 (that **CLB CS)
 (that DET CENTRAL DEM SG)
 (that ADV AD-A)
 (that NonMod PRON DEM SG)
 (that NonMod **CLB Rel PRON SG/PL))
- (very
 (very Altir A ABS)
 (very ADV AD-A))
- (book
 (book N NOM SG)
 (book SVO INF: V SUBJUNCTIVE VFIN)
 (book SVO INF: V IMP VFIN)
 (book SVO INF: V INF)
 (book SVO INF: V PRES -SG3 VFIN))
- (.)

After successful morphological disambiguation, morphosyntactic mapping, and application of syntactic constraints, the final analysis turned out by ENGGC is:¹

- (*her
 (she * NonMod PRON PERS FEM GEN SG3 (@GN>)))
- (hand
 (hand N NOM SG (@SUBJ)))
- (had
 (have INF: SVO SVOC/A V PAST VFIN (@+FALXV)))
- (come
 (come INF: SVC/A SV P/for PCP2 (@-FMMAINV)))
- (to
 (to INFMARK> (@INFMARK>)))
- (rest
 (rest SVC/A SV SVO INF: V INF (@-FMMAINV)))
- (on
 (on PREP (@ADV)))
- (that
 (that DET CENTRAL DEM SG (@DN>)))
- (very
 (very Altir A ABS (@AN>)))
- (book
 (book N NOM SG (@<P)))
- (.)

8.4 Performance levels of ENGGC

Some general observations on the performance of ENGGC are called for. ENGGC contains some 1100 morphological disambiguation constraints developed by Airo Youtainen, and some 400 syntactic

¹ ENGGC has been put at the disposal of the research community for testing purposes. You can easily test the properties of ENGGC, free of charge, by sending an ASCII text file containing no more than 300 word form tokens to the e-mail address:

enggc@ling.helsinki.fi

The parsed text is returned to you as an ordinary e-mail message. Academic users can also buy a User's Licence and thereby obtain a running copy of ENGGC for noncommercial research purposes. Request more details by contacting the present author.

constraints developed by Arto Anttila. The average success levels and error rates to be cited below were determined on the basis of extensive application of ENGGCG to various types of text corpora, followed by subsequent detailed examination of the results.

94.97% of the word-forms of unrestricted input text are fully disambiguated on the morphological level, with an error rate not exceeding 0.3% (Voutilainen, 1993). This compares favourably to CLAWS1 and PARTS, both of which report an error rate around 3-4%. Furthermore, the heuristic options of CG, used on top of the ordinary safe constraints, provide a possibility to discard the remaining ambiguities. This option has a negligible influence on the error rate (Voutilainen, 1993). It will obviously be very hard to improve on these results.

The syntactic constraints are presently less successful. Some 85% of the word-forms end up with a unique syntactic label, with an error rate of 3% (Anttila, 1993). That is, three words out of a hundred have got an inappropriate syntactic label, and fifteen words have more than one label.

Such levels of correctness and precision are reasonably good as a first approximation, cf. the discussion in Section 8.1 of how robustness could be defined. It deserves to be stressed, however, that work on the syntactic constraints has not yet been completed. CG is a modular system and the present set of syntactic constraints is under continuous development.

8.5 Evaluation of the current syntactic performance of ENGGCG

In this section, detailed examples will be provided of how the syntax of ENGGCG fares in the analysis of different types of running text. Genuine example sentences from four different text types are used: a car manual, a novel (*Alice's Adventures in Wonderland*), a computer manual, and neutral descriptive prose (an encyclopedia).

For the purpose of facilitating the ensuing discussion, the words are numbered incrementally. The words will be referred to by their number prefixed by the letter 'w', e.g. w7, w335. The output format seen in the earlier sections is simplified so that for each word only three items are given: the word form, its base form (lemma), and the syntactic label(s) assigned by ENGGCG. All morphological features are omitted.

The underline () is introduced by the preprocessor and marks a compound boundary. Similarly, the equation mark (=) is introduced by the preprocessor and marks the parts of a (more or less) fixed and lexicalized phrase. CG dissociates punctuation marks from their word bodies and transfers them to the following line, prefixed by a dollar sign, e.g. "\$," for the comma.

If a word, after application of ENGGCG, still retains more than one syntactic label, this indicates that the syntactic constraints are not (yet)

precise enough. Such instances are marked by three asterisks (***). Ourlight errors are marked by three exclamation marks (!!!).

(Text 1: Ford Consul Grenada, User's Manual)

1	The	the	@DN>
2	cast_iron	cast_iron	@NN>
3	cylinder	cylinder	@NN>
4	block	block	@SUBJ
5	is	be	@+FMAINV
6	integral	integral	@PCOMPL-S
7	with	with	@<NOM @ADVL ***
8	the	the	@DN>
9	upper	upper	@AN>
10	half	half	@<P
11	of	of	@<NOM-OF
12	the	the	@DN>
13	crankcase	crankcase	@<P
S,			
14	the	the	@DN>
15	lower	low	@AN>
16	half	half	@SUBJ
17	of	of	@<NOM-OF
18	which	which	@<P
19	is	be	@+FAUXV
20	formed	form	@-FMAINV
21	by	by	@ADVL
22	the	the	@DN>
23	pressed	press	@AN>
24	steel	steel	@NN>
25	sump	sump	@<P
8,			
26	The	the	@DN>
27	overhead	overhead	@AN>
28	valves	valve	@SUBJ
29	are	be	@+FAUXV
30	mounted	mount	@-FMAINV
31	at	at	@ADVL
32	a	a	@DN>
33	slight	slight	@AN>
34	angle	angle	@<P
35	across	across	@<NOM @ADVL ***
36	the	the	@DN>
37	cylinder	cylinder	@NN> @<P ***

38	head	head	@+FMMAINV
39	and	head	@<P
40	arc	and	@CC
41	operated	be	@+FAUXV
42	by	operate	@-FMMAINV
43	an	by	@ADV L
44	overhead	an	@DN>
45	camshaft	overhead	@AN>
46	driven	camshaft	@<P
47	by	drive	@<NOM-FMMAINV
48	a	drive	@ADV L
49	reinforced	by	@DN>
50	toothed	reinforce	@AN>
51	rubber	tooth	@-FMMAINV @AN> ***
52	belt	rubber	@OBJ @NN> @<P ***
53	from	belt	@SUBJ @OBJ @<P ***
54	the	from	@<NOM @ADV L ***
55	crankshaft	the	@DN>
S.		crankshaft	@<P
56	A	a	@DN>
57	spring-loaded	spring-loaded	@AN>
58	follower	follower	@SUBJ
59	on	on	@<NOM @ADV L ***
60	the	the	@DN>
61	unloaded	unload	@AN>
62	side	side	@<P
63	of	of	@<NOM-OF
64	the	the	@DN>
65	belt	the	@<P
66	servcs	belt	@+FMMAINV
67	to	serve	@INFMARK>
68	tension	to	@-FMMAINV
69	the	tension	@DN>
70	belt	the	@OBJ
S.		belt	
71	The	the	@DN>
72	crankshaft	crankshaft	@SUBJ
73	runs	run	@+FMMAINV
74	in	in	@ADV L
75	five	five	@ON>
76	steel-backed	steel-backed	@AN>
77	shell	shell	@NN>
78	bearings	bear	@<P

79	in	in	@<NOM @ADV L ***
80	the	the	@DN>
81	crankcase	crankcase	@NN>
82	main	main	@AN>
83	journals	journal	@<P
S.		journal	

(Text 2: Lewis Carroll, Alice's Adventures in Wonderland)

84	Alice	alice	@SUBJ
85	was	bc	@+FMMAINV
86	beginning	begin	@+FAUXV ***
87	to	to	@-FMMAINV
88	get	get	@INFMARK>
89	very	very	@-FMMAINV
90	tired	tire	@AD-A>
91	of	of	@AN>
92	sitting	of	@ADV L
93	by	sit	@<P-FMMAINV
94	her	by	@<NOM @ADV L ***
95	sister	she	@GN>
96	on	sister	@<P
97	the	on	@<NOM @ADV L ***
98	bank	the	@DN>
S.		bank	@<P
99	and	and	@CC
100	of	of	@<NOM-OF
101	having	have	@<P-FMMAINV
102	nothing	nothing	@SUBJ @OBJ
103	to	to	@INFMARK>
104	do	do	@-FMMAINV
S.		do	
105	once=or=twice	once=or=twice	@ADV L
106	she	she	@SUBJ
107	had	have	@FAUXV
108	peeped	peep	@-FMMAINV
109	into	into	@ADV L
110	the	the	@DN>
111	book	book	@<P
112	her	she	@GN>
113	sister	sister	@O-ADV L
114	was	bc	@+FAUXV
115	reading	read	@-FMMAINV

\$,	116 but	but	@CC
\$,	117 it	it	@SUBJ
\$,	118 had	have	@+FMMAINV
\$,	119 no	no	@QN>
\$,	120 pictures	picture	@OBJ
\$,	121 or	or	@CC
\$,	122 conversations	conversation	@SUBJ @OBJ
\$,	123 in	in	@ADVL
\$,	124 it	it	@<P
\$,	125 and	and	@CC
\$,	126 what	what	@SUBJ
\$,	127 is	be	@+FMMAINV
\$,	128 the	the	@DN>
\$,	129 use	use	@PCOMPL-S
\$,	130 of	of	@NOM-OF
\$,	131 a	a	@DN>
\$,	132 book	book	@<P
\$,	133 thought	think	@APP @-FMMAINV
\$,	134 Alice	think thought alice	@AN> @+FMMAINV @APP @NN> @SUBJ @OBJ @PCOMPL-O @APP @<P
\$,	135 without	without	@ADVL
\$,	136 pictures	picture	@<P
\$,	137 or	or	@CC
\$,	138 conversation	conversation	@OBJ @<P
\$,	139 In=	in=	@ADVL
\$,	140 running	run	@NN @P-FMMAINV
\$,	141 programs	program	@AN>
\$,	142 from	from	@SUBJ @OBJ @<P*** @<NOM @ADVL ***

(Text 3: OS/2 Computer Manual)

\$,	143 the	the	@DN>
\$,	144 DOS	dos	@NN>
\$,	145 Command	command	@NN>
\$,	146 Prompt	prompt	@<P
\$,	147 that	that	@SUBJ
\$,	148 is	be	@+FAUXV
\$,	149 provided	provide	@-FMMAINV
\$,	150 in	in	@ADVL
\$,	151 the	the	@DN>
\$,	152 OS/2	os/2	@NN> @<P
\$,	153 operating	operate	@NN
\$,	154 system	system	@<NOM-FMMAINV
\$,	155 you	you	@-FMMAINV @AN>***
\$,	156 can	can	@OBJ @<P
\$,	157 set	set	@OBJ @<P
\$,	158 up	up	@ADVL
\$,	159 your	you	@GN>
\$,	160 machine	machine	@OBJ @I-OBJ
\$,	161 to	to	@INFMARK>
\$,	162 run	run	@-FMMAINV
\$,	163 a	a	@<NOM-FMMAINV ***
\$,	164 version	version	@DN>
\$,	165 of	of	@OBJ
\$,	166 the	the	@<NOM-OF
\$,	167 DOS	dos	@DN>
\$,	168 operating	operate	@NN>
\$,	169 system	system	@NN>
\$,	170 on	on	@<P
\$,	171 the	the	@<NOM @ADVL ***
\$,	172 same	same	@DN>
\$,	173 system	system	@AN>
\$,	174 as	as	@<P
\$,	175 OS/2	os/2	@ADVL
\$,	176 Some	some	@<P
\$,	177 DOS	dos	@QN>
\$,	178 programs	program	@NN>
\$,	179 may	may	@SUBJ
\$,	180 not	not	@+FAUXV
\$,	181 run	run	@NEG @-FMMAINV

182	under	under	@ADV L	***
183	OS/2	os/2	@NN @<P	***
184	Standard	standard	@NN> @<P	***
185	Edition	edition	@NN> @<P	***
186	Version	version	@NN>	***
187	1.3	1.3	@<P	
S.				
188	If	if	@CS	
189	you	you	@SUBJ	
190	need	need	@+FM AIN V	
191	to	to	@IN F MARK >	!!!
192	run	run	@-FM AIN V	
193	DOS	dos	@NN>	
194	programs	program	@OBJ	
195	that	that	@SUBJ	
196	are	be	@+FM AIN V	
197	time-dependent	time-dependent	@PCOM PL-S	
S(
198	such=as	such=as	@ADV L	
199	communication	communication	@NN> @<P	***
200	and	and	@CC	
201	real-time	real-time	@NN>	
202	programs	program	@<P	
S)				
203	or	or	@CC	
204	hardware-dependent	hardware-dependent	@SUBJ @<P	!!!
S(
205	such=as	such=as	@ADV L	
206	networking	networking	@NN> @<P	***
207	and	and	@<P-FM AIN V	
208	driver	driver	@CC	
209	programs	program	@NN> @<P	***
S)			@SUBJ @OBJ @<P***	
\$,				
210	sec	sec	@+FM AIN V	
			@-FM AIN V	
			@<P-FM AIN V !!!	***
211	A.2	a.2	@OBJ	
\$,				
\$*				
212	Using	use	@OBJ @PCOM PL-O	
			@APP @NN	

213	Dual	dual	@<P @+FM AIN V	***
214	Boot	boot	@AN>	***
\$*			@OBJ @APP @<P	***
215	on	on	@ADV L	
216	page	page	@<P	
217	195	195	@SUBJ	!!!
218	before	before	@ADV L	
219	installing	instal	@<P-FM AIN V	
220	the	the	@DN>	
221	OS/2	os/2	@OBJ @NN>	***
222	operating	operate	@NN>	
			@<NOM-FM AIN V @-	
			FM AIN V @AN>	***
			@OBJ	
223	system	system		
S.				

(Text 4: Grolier International Encyclopedia, "World War I")

224	The	the	@DN>	
225	assassination	assassination	@O-ADV L	!!!
226	of	of	@<NOM-OF	
227	the	the	@DN>	
228	Austrian	austrian	@AN>	
229	archduke	archduke	@N	
230	Franz	franz	@N	
231	Ferdinand	ferdinand	@<P	
232	in	in	@ADV L	
233	Sarajevo	sarajevo	@<P	
234	in	in	@<NOM @ADV L	***
235	1914	1914	@SUBJ	!!!
236	proved	prove	@+FM AIN V	
237	to	to	@IN F MARK >	
238	be	be	@-FAUX V	
			@-FM AIN V	***
239	the	the	@DN>	
240	spark	spark	@PCOM PL-S	
241	that	that	@SUBJ	
242	ignited	ignite	@+FM AIN V	
243	World_War_I	world_war_i	@OBJ	
S(
244	1914-18	1914-18	@NPHR	
S)				

\$,	245 Called	call	@AN>	!!!
\$"	246 the	the	@DN>	
	247 Great	great	@AN>	
	248 War	war	@OBJ	
\$,				
	249 it	it	@SUBJ	
	250 quickly	quick	@ADVL	
	251 came	come	@+FMMAINV	
	252 to	to	@INFMARK>	
	253 involve	involve	@-FMMAINV	
	254 all	all	@QN>	
	255 the	the	@DN>	
	256 great	great	@AN>	
	257 powers	power	@OBJ	
	258 of	of	@<NOM-OF	
	259 Europe	europ	@<P	
	260 and	and	@CC	
	261 eventually	eventual	@ADVL @AD-A>	***
	262 most	many	@QN>	
	263 countries	country	@SUBJ @OBJ @<P***	
	264 of	of	@<NOM-OF	
	265 the	the	@DN>	
	266 world	world	@<P	
\$,				
	267 and	and	@CC	
	268 cost	cost	@+FMMAINV	
		cost	@-FMMAINV	
	269 the	the	@DN>	
	270 lives	life	@OBJ @APP	***
	271 of	of	@<NOM-OF	
	272 more=than	more=than	@ADVL @AD-A>	***
	273 8	8	@<P @QN>	***
	274 million	million	@QN>	
	275 soldiers	soldier	@<P	
\$,				
	276 Among	among	@ADVL	
	277 the	the	@DN>	
	278 causes	cause	@<P	
	279 of	of	@<NOM-OF	
	280 the	the	@DN>	
	281 war	war	@<P	

	282 were	bc	@+FMMAINV	***
	283 rising	rise	@+FAUXV	***
			@NN> @-FMMAINV	***
	284 nationalist	nationalist	@AN>	
	285 sentiment	sentiment	@NN>	
			@SUBJ	
\$ (@PCOMPL-S	***
	286 manifested	manifest	@APP @-FMMAINV	***
	287 both	both	@AN>	
		both	@CC	
	288 in	in	@OBJ	!!!
	289 the	the	@ADVL>	
	290 chauvinism	chauvinism	@DN>	
	291 of	of	@<P	
	292 the	the	@<NOM-OF	
	293 great	great	@DN>	
	294 European	europ	@AN>	
	295 powers	power	@AN>	
	296 and	and	@<P	
	297 in	in	@CC	
	298 the	the	@ADVL	
	299 unrest	unrest	@DN>	
	300 among	among	@<P	
	301 the	the	@<NOM @ADVL	***
	302 subject	subject	@DN>	
	303 peoples	people	@NN>	
	304 of	of	@<P	
	305 the	the	@<NOM-OF	
	306 multinational	multinational	@DN>	
	307 European	europ	@AN>	
	308 empires	empire	@AN>	
\$)			@<P	
\$,				
	309 colonial	colonial	@AN>	
	310 and	and	@CC	
	311 economic	economic	@AN>	
	312 rivalries	rivalry	@SUBJ @OBJ @APP	***
			@<P	
\$,				
	313 the	the	@DN>	
	314 formation	formation	@SUBJ @OBJ @APP	***
			@<P	

315 of	of	@<NOM-OF
316 hostile	hostile	@AN>
317 alliance	alliance	@NN>
318 systems	system	@<P
\$,		
319 and	and	@CC
320 arms_races	arms_race	@SUBJ @OBJ @APP
		@<P

S,		
321 all	all	@SUBJ
322 of	of	@<NOM-OF
323 which	which	@<P
324 contributed	contribute	@+FMAINV
325 to	to	@ADVL
326 the	the	@DN>
327 growing	grow	@NN> @AN>
328 sense	sense	@<P
329 of	of	@<NOM-OF
330 international	international	@AN>
331 tension	tension	@<P
332 during	during	@<NOM @ADVL ***
333 the	the	@DN>
334 prewar	prewar	@AN>
335 years	year	@<P
\$)		

There are 335 word form tokens in the sample. The morphological analyser properly analysed 324 words (96.7%). Eight of the remaining eleven words unknown to the morphological analyser were instances of the lexemes *DOS* and *OS/2*. The heuristic part-of-speech assignment rules of ENCGG were successful in predicting the part of speech of these which therefore also ended up with appropriate syntactic labels (e.g. w144, w167, w177, w152, w175, w183). No errors were committed due to morphological analysis.

329 words (98.2%) are unambiguous after morphological disambiguation. Six morphological ambiguities remain, the words w38 *head*, w133 *thought*, w206 *networking*, w210 *see*, w268 *cost*, and w287 *both*. Half of these relate to noun/verb ambiguities, a pervasive problem in the parsing of English. It is notable that most of the 'hesitation' of the disambiguation module is due either to occurrence in a conjoined structure (w38, w206, w268), or in a marked word order configuration (w133, w210). These are notorious syntactic problems. It is, in fact, an indication of proper regimentation among the disambiguation constraints that such difficult

and risky situations are clearly spotted. At the discretion of the constraint writer, heuristic constraints could be invoked.

Even the six remaining ambiguous cohorts have been reduced very close to optimal size. All contain two readings except for *see* with three. No heuristic disambiguation constraints were used.

There are nine syntactic errors (2.7%). W90 *tired* is not a premodifier (@AN>) but rather a subject predicate complement, and w91 *of* is not the head of a top level adverbial but rather a complement of the preceding adjective *tired* whose complementation properties are not adequately treated. The current constraints for object adverbials (@O-ADVL, intended for phrases like *mile* in e.g. *run a mile*) are clearly inadequate, as witnessed by the embarrassing errors w113 and w225. In w204, which occurs in a very complicated coordinated construction, the appropriate label @PCOMPL-S predicate complement of the subject, has been discarded and only two improper labels remain (@SUBJ, @<P).

W217, a numeral, is inadequately claimed to be @SUBJ where the correct label (in the present framework) would be postmodifier (in the expression *page 195*). In a similar way, the numeral w235 (1914) is claimed to be subject rather than complement of the immediately preceding proposition. The constraints for cardinal number expressions obviously should be revised. W245 *called* is erroneously claimed to be premodifier when the correct label would be non-finite predicate. This error resembles that of w90. Finally, the second reading remaining for 287 *both* (pronoun) is given the improper label @OBJ (this reading should be discarded).

Generalizing over the errors, it is notable that some of them fall into clearly discernible groups giving hints to the constraint grammar writer as to what (types of) syntactic constraints need elaboration and correction. For instance, the constraints for cardinal numbers, object adverbials, adjectival premodifiers, and coordinated constructions are not fully acceptable yet.

57 words (17%) remain syntactically ambiguous, e.g. w7, w35, w50, w51, w52, w53, w59. Note how attachment ambiguities are conveniently represented. For example, in w7 (*with*), the label @<NOM represents postmodification (low attachment to the preceding head) whereas the label @ADVL represents high attachment under the main verb.

Thus the overall morphological and syntactic performance of ENCGG in regard to the present small corpus is within the quality limits reported by Youtilainen and Anttila (section 8.4), disregarding the 2% drop in syntactic precision (83% rather than 85% unambiguous).

Due to the overall modularity of CG, and to the concreteness of the individual constraints, even large constraint grammars tend to be manageable in the sense that it is possible to go on with testing and refinements of the extant constraints, and with addition of new ones,

almost ad infinitum. Therefore the current performance limits of ENGCG are not conclusive.

Work on improving the analytical behaviour of ENGCG goes on perpetually. In 1992, a contract was signed between the Research Unit for Computational Linguistics at the University of Helsinki (RUCL) and HarperCollins Publishers, Glasgow, concerning tagging the whole prospective Bank of English with ENGCG part-of-speech and syntactic codes. The Bank of English is a 200-million-word text corpus conceived by John Sinclair and his colleagues at COBUILD, University of Birmingham. RUCL undertook to do the tagging, by way of using the ENGCG program, at a pace of 10 million words per month, starting in February 1993. The whole corpus will thus have been tagged by the end of 1994. Presently (September 1993), 80 million words have been tagged and sent to Birmingham. Such a huge corpus project provides excellent opportunities for testing and updating all modules of the system. Special attention is presently paid to improving the syntactic surface analysis capacity of ENGCG, as well as to updating the ENGTWOL master lexicon that constitutes the lexico-morphological kernel of the system.

8.6 A note on processing time

All of the following processing figures pertain to the performance of ENGCG on a Sun SparcStation 10 (Model 30) where precisely the same lexicon and constraint files were used by the different programs. The Lisp version of CGP is normally used for development and testing of a constraint grammar. It does full syntactic parsing of input text at an average speed of 3-5 words per second, regardless of the structural complexity of the input sentences. The C++ version of CGP programmed by Bart Jongejan parses at a level of 15-20 words per second, also regardless of input sentence complexity. Recently, Pasi Tapanainen has completed a new optimized C++ version of CGP parsing running English text at a pace of 400-500 words per second (morphological analysis, disambiguation, and syntactic analysis of the above type included).