

Workshop Notes from the Ninth  
National Conference on  
Artificial Intelligence

(AAAI-91)

Natural Language Text  
Retrieval

American Association for Artificial Intelligence

July, 15 1991

Anaheim, California

**AAAI-91**  
**Natural Language Text Retrieval Workshop**

Monday, July 15, 1991  
Anaheim Convention Center, Anaheim, CA

**American Association of Artificial Intelligence**

Fred Karlsson, Atro Voutilainen, Arto Anttila, Juha Heikkilä

Constraint Grammar (CG) is a language-independent formalism for surface-oriented, morphology-based parsing of unrestricted text. Descriptions exist for English (henceforth, ENGCG), Swedish, and Finnish. CG parsing is based on the disambiguation of (i) morphological readings and (ii) syntactic codes. All relevant structure is assigned directly via the lexicon, morphology, and simple mappings from morphology to syntax. The constraints discard as many contextually illegitimate alternatives as possible. A comparison between ENGCG and two stochastic disambiguators is reported. ENGCG is used for large-scale text indexing in the ESPRIT II project SIMPR (Project 2083: *Structured Information Management: Processing and Retrieval*).

## 1. Aims of Constraint Grammar

There are two basic types of parsers. One is based on autonomous grammar models such as GB, LFG, or GPSG, e.g. Briscoe, Grover, Boguraev, and Carroll (1987). The other type uses stochastic methods, e.g. CLAWS (see Garside, Leech, and Sampson 1987), or Church's (1988) parts and noun phrase program for English.

Constraint Grammar (Karlsson 1990) mediates between these two approaches. Compared to parsers incorporating autonomous grammar models, CG is similar in operating with rule-like statements, called constraints, that employ traditional linguistic categories, especially parts of speech. CG is different from these models, however, as it pays much attention to morphology as the basis for syntactic analysis, to real syntactic properties of text sentences, and to some notorious parsing problems, especially ambiguity. A major difference is that CG constraints always eliminate alternatives judged to be incorrect in a certain context. The constraints thus do not have the ordinary task of defining the notion 'correct sentence in language L', even if they rely on the fact that most sentences ordinarily are correct. Constraints capitalize on the surface structure redundancies ultimately due to language rules.

Compared to purely stochastic approaches, CG shares the concern with large corpora and the attempt not to let infrequent structural possibilities unduly complicate the grammar and the parsing process. But CG still is decidedly 'qualitative' rather than probabilistic in nature. Individual constraints use well-known linguistic categories and it is left to the discretion of the CG writer to judge how 'risky' the constraints should be. A major difference is that CG, both the formalism and the present implementation, is applicable as such to any language.

The input to the CG parser consists of properly preprocessed and lexically analysed text, for example:

```
it * <NonMod> PRON NOM SG3 SUBJ @SUBJ *
it * <NonMod> PRON ACC SG3 * <<<
```

```
cost
cost * <SVO> <SVOO> PCP2 * <<<
cost * <SVO> <SVOO> V PAST VFIN @+FMAINV *
cost * <SVO> <SVOO> V SUBJUNCTIVE VFIN @+FMAINV * <<<
cost * <SVO> <SVOO> V IMP VFIN @+FMAINV * <<<
cost * <SVO> <SVOO> V INF * <<<
cost * <SVO> <SVOO> V PRES -SG3 VFIN @+FMAINV * <<<
cost * N NOM SG * <<<
```

```
those
that * <NonMod> PRON DEM PL * <<<
that * DET CENTRAL DEM PL @ON * <<<
```

```
people
people * N NOM SG/PL *
people * <SVO> <P/with> <Rare> V IMP VFIN @+FMAINV * <<<
people * <SVO> <P/with> <Rare> V INF * <<<
```

```
several
several * <NonMod> PRON NOM PL * <<<
several * <Quant> DET POST PL @QN * <<<
```

```
pounds
pound * <SVO> <SV> V PRES SG3 VFIN @+FMAINV * <<<
pound * N NOM PL * <<<
```

s.

(Readings manually marked with '<<<' were discarded by the disambiguation module.) For each word-form, a base-form and a string of tags is given. Each word-form gets one or more analyses, together forming a cohort.

Optimally, CG should assign each word-form token only contextually justified morphological readings and surface syntactic codes indicating the function of the word in the clause. CG uses functional dependency syntax, i.e., for each word it is indicated whether it is a head or a modifier, and what type of head or modifier it is. The structures assigned are flat. The output is easy to examine and postprocess.

## 2. Constraint Grammar Design

Full-fledged CG parsing presupposes four robust processing modules: preprocessing of raw text (conversion to lower case, identification of sentence delimiters, and detection of idioms), morphological analysis (using Koskenniemi's (1983) two-level model), local disambiguation (eliminating e.g. spurious morphological readings possibly arising in the analysis of compounds), and Constraint Grammar Parsing, identifying intrasentential clause boundaries and performing two types of disambiguation, lexical and syntactic. Our description of English presupposes no local disambiguation.

The basic type of constraint is a quadruple consisting of a domain, operator, target, and context conditions. The domain points out some element to be disambiguated, e.g. a particular word-form or reading. The operator states what operation to perform, e.g. discard a particular reading or code, leaving the other ones intact, or pick a particular reading or code as uniquely correct, discarding all the other ones. The target defines which reading the constraint applies to if there are several candidates. The context conditions refer to surrounding words as needed, including possibilities for covering long-distance dependencies. For instance, the disambiguation constraint

```
(@w =0 V (-2 DET) (-1 ADV))
```

states that all verb readings in the cohort are to be discarded if the immediately preceding word is an adverb and the next word to its left a determiner.

Our English Constraint Grammar ENGCG presently contains approximately 1,100 disambiguation constraints written by Atro Voutilainen (see Section 4), and more than 400 syntactic constraints, written by Arto Anttila (see Section 5). In conjunction with the morphological analyzer, compiled by Atro Voutilainen and Juha Heikkilä (see Section 3), the Constraint Grammar Parser yields output like:

```

"i
* " <NonMod> PRON NOM SG3 SUBJ @SUBJ *
cost
cost " <SV> <SVO> <SVOO> V PAST VFIN @+PMAINV *
those
that " DET CENTRAL DEM PL @DN *
people
people " N NOM SG/PL " @I-OBJ
several
several " <Quant> DET POST PL @QN *
pounds
pound " N NOM PL " @OBJ
s

```

### 3. Lexicon and Morphological Analyser

In ENGCG, the lexicon and the morphological analyser, ENGTWOL, have been designed according to Koskenniemi's (1983) two-level model. At present, the lexicon contains some 52,000 entries. These include a little under 600 idioms and about 5,400 compounds, the majority of which were extracted from the *Collins COBUILD English Language Dictionary*. Idioms and compounds are detected and marked specially during the preprocessing of the input text.

The compilation of the lexicon was carried out by making extensive use of two corpora, the Brown University Corpus and the London-Oslo/Bergen Corpus (LOB). Other sources in electronic form include *The New Grolier Electronic Encyclopedia* and issues of *The Wall Street Journal* (courtesy of the ACL Data Collection Initiative). Furthermore, the *Longman Dictionary of Contemporary English* was manually checked for missing words to ensure that the lexicon adequately covers the core vocabulary of English. One of the aims in compilation has been to make the lexicon variant-independent - in the present form it covers the two major variants of English, British and American.

For each input word-form, the morphological analyser returns a cohort of readings each of which consists of the base form and a string of features. These features indicate the part of speech of the word-form and its inflectional and other morphosyntactic properties. The whole CG description of English constitutes an integrated system in which the different levels of analysis have been closely co-ordinated. Thus, the lexicon has been expressly designed to meet the requirements of disambiguation and syntax; the principle of feature assignment has been to offer the later stages of analysis as much information as possible as early as possible, providing the linguist with a useful set of tools. Consequently, systematic categorical ambiguity has been allowed, instead of setting up underspecified categories. Furthermore, new features (e.g., transitivity features) have been added to the lexicon once the need has arisen. The current total number of features is 159, consisting of morphosyntactic (including part of speech), derivational and stylistic features. The part-of-speech categories are roughly based on the classification of Quirk, Greenbaum, Leech & Svartvik (1985). With this feature set

and classification, about 45-50% of running-text word-forms are at least two-ways ambiguous.

The coverage of the lexicon (*recall*) was measured by running ENGTWOL in the filtering mode, which prints out all word-forms not recognised by the analyser. The test was conducted with three previously unprocessed texts: the first of these, about 10,000 words, was made up of articles from *The New Grolier Electronic Encyclopedia*; the second was an 11,000-word extract from *The Wall Street Journal*; the third text consisted of 14,000 words from a computer manual. Because of the similarity of the results produced by the first two texts, these will be reported together.

In the first two texts, some 3% of the word-form tokens, 2.7% of the word-form types, were not recognised by the morphological analyser. About 78% of these were names and abbreviations. Overall, almost all unrecognised word-forms were nominals - only two verbs were encountered. About 10% of the unrecognised words might be included in the lexicon, these being mainly compounds.

As for the computer-manual text, about 5.5% of the tokens, 1.5% of the types, were not recognised by ENGTWOL. These were all nominals, of which 85% abbreviations. The percentage of unrecognised word-form tokens in the second test set was higher than usual. Spelling errors occurred in both test sets, and they represented 0.1% of the word-form tokens.

Names and abbreviations constitute a vast majority of the unrecognised word-forms in these and in most other texts. The number of compounds is also large. Naturally, common compounds are included in the lexicon; however, their formation is a productive process in English, and texts often contain a number of ad hoc formations in premodifying structures. These remain unrecognised because the lexicon system does not allow recursion back to the stem lexicon. On the whole, the filtered items clearly fall into these three major groups whose members are more or less text-specific. This indicates that the coverage of the lexicon is satisfactory. Moreover, the heavy part-of-speech concentration of the unrecognised items suggests that the updating of the lexicon could be automated with reasonable accuracy. The correctness rate of ENGTWOL, i.e., correctness of the recognised items, is extremely close to 100%.

### 4. Lexical Disambiguation

At the morphosyntactic level of description, ENGTWOL accounts for all and only the legitimate uses of each recognised word-form. The assignment of more than one analysis signals that the word-form in question has more than one use. Lexical analysis is not sensitive to syntactic context.

Generally, only one of the analyses in a cohort is contextually legitimate. The task of lexical disambiguation is thus to discard all and only the contextually illegitimate uses.

Consider the sample input presented in Section 1. An actual run discarded all readings marked with '<<<'. To paraphrase some of the activated constraints, we could state that the accusative case (ACC) is legitimate only if there is a transitive verb (<SVO>, <SVOO>) or a preposition (PREP) in an appropriate position; imperatives (IMP) do not occur in clauses containing a subject (@SUBJ); infinitives (INF) are preceded by certain kinds of words (*to* or a modal auxiliary like *may* or certain lexical verbs like *see*), etc.

The present ENGCG disambiguation grammar contains some 1,100 carefully tested constraints embody-

ing essentially negative statements, much like the few cited above. At present, some 94–97% of all words become fully disambiguated (*recall*), and of all words, no more than 0.3% miss the most appropriate analysis (*precision*). The grammar is still under development, because it seems possible to reach a higher recall – perhaps 96% to 98% – without compromising precision.

So far, the most powerful publicly-known part-of-speech disambiguators of English are based on probabilistic techniques. Probabilities based on frequencies of (strings of) words and tags are counted, and the readings with the highest score are selected as the most likely analyses. CLAWS1 by the UCREL group (Garside, Leech & Sampson 1987) and PARTS by Church (1988) are probably the best in the field. With 100% recall, the precision is reported to be 96–97% for CLAWS, and 95–99% for PARTS.

A test-based comparison between the three disambiguators is made in the remainder of this section. A comparison presupposes, however, (i) some evaluation of the granularity of the three descriptions (the more fine-grained the description, the more challenging the task of disambiguation), and (ii) some understanding of the respective problems posed by leaving either ambiguity or errors in the output.

The descriptions used in the three systems are not identical. The differences do not, however, appear radical. The feature system in CLAWS (the tag set used in the tagging of LOB corpus) is clearly more fine-grained than that used in PARTS (largely that used in the Brown Corpus). The CG feature system; on the other hand, is more detailed in some respects than CLAWS. To cite the most prominent differences, the CLAWS tag system makes no distinction between pronoun-determiner readings in words like *this*, whereas ENGTWOL does; ENGTWOL has four different verb readings for what is represented as VB in the other systems; certain adjective-pronoun-adverb homographs are spelt out as different readings in ENGTWOL, and left underspecified in CLAWS. On the other hand, ENGTWOL does not make a difference between the nominal and the verbal readings of participles, which CLAWS does. Overall, we may conclude that the ENGTWOL tag system is somewhat more fine-grained than that used in CLAWS; both are more fine-grained than that used in PARTS.

As regards the second problem, our understanding is that an error rate of, say, more than one per cent can be considered harmful for practical purposes (see Johansson, Atwell, Garside, Leech 1986:21). Therefore, a system producing an unacceptable amount of errors presupposes comprehensive postediting. This is at least partly manual, and the look-up and correction of errors through the whole text – as can be the case with probabilistic systems – is a time-consuming effort (Johansson & al. 1986:21). Therefore it seems practical to minimise the amount of manual work. The present disambiguation grammar is designed precisely with this goal in mind. While maximal recall is seen as desirable, some ambiguity in the output is tolerated, if there is no other way to avoid a considerable risk of error. Thus the result is that whatever is fully disambiguated is probably also correctly disambiguated, and needs no further post-editing. (It is equally likely that whatever remains ambiguous, also contains the most appropriate reading.) Therefore all that remains to be done after grammar-based disambiguation is to process the remaining, easily detectable ambiguities – by manual postediting, heuristics, or whatever.

Johansson & al. (1986:21) reports that CLAWS

can disambiguate 86% of all text fully with a 1% error rate (thus 14% of all text remains for manual postediting – given that an error rate of 1% is tolerated). The present version of ENGCG disambiguation is consistently capable of disambiguating 94–97% of all fairly unmarked English prose text without exceeding an error rate of 0.3%.

We now present a preliminary report on the performance of these three systems (for a full account, see Karlsson, Voutilainen, Heikkilä, Anttila, forthcoming). Three roughly equal-sized texts, new to ENGCG, were chosen for the test run. One was a recent theatre review from a newspaper, the second, an excerpt from a book on the philosophy of language, the third, an extract from a software manual (total 1329 words).

Here are the results:

	Recall	Precision
(1) CLAWS1	100%	96.32% (49 errors)
(2) PARTS	100%	96.01% (53 errors)
(3) ENGCG	96.4%	99.70% (4 errors)

Roughly speaking, the performance of CLAWS and PARTS was equally good. The errors were mainly of the following kinds: [preposition / subordinating conjunction], [adjective / noun], [noun / verb], [past tense / past participle], and [preposition / adverb]. (There were some uncertain cases as well, but, due to uncertainty on the part of the examiner, they were not counted as errors.)

The four errors committed by ENGCG are listed here:

- .. being open to[INFMARK, SHOULD BE PREP]
- doubt[INF, SHOULD BE N] does not render ..
- .. the save[V/PREP, SHOULD BE N] menu-item is only enabled ..
- .. the autobiography of a Japanese woman, Yoshiko Yamaguchi, "Ri Koran Watashi no[DET, SHOULD BE ??] Hansei" ..

The first two errors result from the absence of the complementation properties from the lexical representation for the adjective *open* (both other systems committed the same error as well); the third can be considered meta-linguistic usage, it could perhaps be avoided if typographical changes were recognised by the preprocessing module; the fourth is not English at all. – Overall, the recall of ENGCG was somewhat towards the upper end of its normal recall of 94–97%, but, on the other hand, precision was lower than the ordinary 99.8–99.9%.

To conclude, the recent success of probabilistic NLP systems has cast some doubt on the feasibility of grammar-based systems in the analysis of unconstrained text (see e.g. Garside, Leech & Sampson 1987). Our experiences show, however, that the grammar-based approach is not doomed to failure. Given the criteria set above, we are unaware of any system (probabilistic or other) that matches the performance of the grammar-driven ENGCG disambiguation system presented above.

## 5. The Syntactic Module

We shall now demonstrate how the different components of a Constraint Grammar contribute to syntactic parsing. Consider the sentence *She gave him an apple pie*. In the resulting parse, each word carries a dependency-oriented syntactic tag, marked by the character '@'.

\*she  
 \* she \* PRON PERS FEM NOM SG3 @SUBJ \*

gave  
 gave \* <SVO> <SVOO> <SV> V PAST VFEN @+FMAINV \*

him  
 he \* <NonMod> PRON PERS MASC ACC SG3 \* @I-OBJ

an  
 an \* <Indef> DET CENTRAL ART SG @DN \*

apple  
 apple \* N NOM SG \* @NN >

pie  
 pie \* N NOM SG \* @OBJ

5.

The forms *she*, *gave* and *an* are the easy cases. For example, we know that the form *she* will always be Subject, hence the tag @SUBJ may be included in its lexical entry. Thus, making *she* @SUBJ is conceptually part of the lexical lookup and no further parsing is needed. The same holds for *gave* (@+FMAINV = Finite Main Verb) and *an* (@DN = Determiner).

On the other hand, *him*, *apple* and *pie* need parsing, as out of context they are syntactically ambiguous. *Him* will initially receive two syntactic tags, @OBJ (= Object) and @I-OBJ (= Indirect Object), as it may be either. *Apple* and *pie* are even worse; given no context, at least the following alternatives must be considered: @SUBJ, @OBJ, @I-OBJ, @PCOMPL-S (= Subject Complement, as in *This is an apple*), @PCOMPL-O (Object Complement, as in *I thought it an apple*), @NN> (=Nominal Premodifier, here the correct reading), @<P (= Preposition Complement, as in *The colour of an apple*), and a few others. What is needed is a module which will assign each form all the theoretically possible syntactic tags. The module is called MORPHOSYNTACTIC-MAPPING, and it consists of a set of (possibly context-sensitive) mappings from morphological tags or word forms onto sets of syntactic functions.

The constraints state the possibilities of occurrence for each syntactic function. In procedural terms, they prune the superfluous syntactic tags, leaving the contextually appropriate one(s) intact. First consider *pie*. The constraint blocking the Subject reading implements the following strategy: "If there is no Finite Verb to the left and no Auxiliary Verb to the right, block @SUBJ." Given a few more rules, the tag @OBJ will be the only one to survive. Second, the fact that the Object of the Main Verb has now been recognised implies that all the other @OBJ-tags in the clause are spurious and can be discarded. This parsing strategy, called the Uniqueness Principle, applies to certain HEAD-functions, such as @SUBJ and @OBJ. Third, *apple* will be made unambiguous by the application of one constraint which will block all functions except @NN>. This constraint relies on the presence of the immediately preceding indefinite article *an* and the immediately following noun *pie*.

From the resulting parse, one is able to spot the HEAD-functions, such as @SUBJ and @OBJ, which typically represent the most prominent discourse entities. One also gets MODIFIER-functions, such as @NN>, which depend on their respective heads and modify them semantically. One may also reconstruct phrases on the basis of the syntactic tagging. As MODIFIERS, such as @NN> and @DN>, also indicate the direction where their HEAD is to be found, a low-level phrase structure is implicitly present.

A test run on fresh data (see Section 4) gave the following results. (Recall here means the percentage of syntactically unambiguous word-forms;

precision stands for error rate.)

	Recall	Precision
(1) Computer manual	89.6%	3.6%
(2) Philosophy reader	84.8%	2.9%
(3) Theatre review	82.2%	3.4%

Three remarks are in order. As the parsing modules operate serially, syntax being the last phase, any error in the previous stages of analysis will result in a syntactic misparse. Second, some of the remaining ambiguity is genuine, PP-attachment and coordination being well-known examples. Third, the work on syntax is still in progress and one may expect substantial improvement in the near future.

## 6. Implementation and Efficiency

The CG parser is implemented in Common Lisp and runs on several types of workstation. An implementation in C is under development. ENGCG is fully operational on unedited text. On a Sun SPARC2 workstation, the present parsing speed is around 3 words per second, including morphology and full syntax.

## References

- Briscoe, T., Grover, C., Boguraev, B. & Carroll, J. 1987. "A Formalism and Environment for the Development of a Large Grammar of English." *IJCAI-87*, Vol. 2, pp. 703-708.
- Church, K. 1988. "A Stochastic Parts Program and Noun Phrase Parser for Running Text." *Second Conference on Applied Natural Language Processing*, ACL 1988, pp. 136-143.
- Garside, R., Leech, G. & Sampson, G. (eds.) 1987. *The Computational Analysis of English: A Corpus-Based Approach*. Longman: London and New York.
- Johansson, S., with E. Atwell, R. Garside, G. Leech 1985. *The Tagged LOB Corpus: User's Manual*. Norwegian Computing Centre for the Humanities, Bergen.
- Karlssohn, F. 1990. "Constraint Grammar as a Framework for Parsing Running Text". In H. Karlgren, ed., *Papers presented to the 13th International Conference on Computational Linguistics*, Helsinki 1990, Vol. 3, pp. 168-173.
- Karlssohn, F., Voutilainen, A., Heikkilä, J. & Anttila, A. (forthcoming). *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*.
- Koskenniemi, K. 1983. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Publications No. 11, Department of General Linguistics, University of Helsinki.
- Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J. 1985. *A Comprehensive Grammar of the English Language*. London: Longman.

Address of the authors

Department of General Linguistics  
 University of Helsinki  
 Hallituskatu 11, SF-00100 Helsinki, FINLAND

karlssohn@finuh.bitnet, phone: +358 0 1913512  
 voutilainen@finuh.bitnet 1913503  
 jheikkila@finuh.bitnet 1913503  
 aanttila@finuh.bitnet 1913500  
 fax: +358 0 653726