Developing a hybrid NP parser

Atro Voutilainen
Department of General Linguistics
P.O. Box 4
FIN-00014 University of Helsinki
Finland
avoutila@ling.helsinki.fi

Lluís Padró
Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
C/ Gran Capità s/n. 08034 Barcelona
Catalonia
padro@lsi.upc.es

Abstract

We describe the use of energy function optimisation in very shallow syntactic parsing. The approach can use linguistic rules and corpus-based statistics, so the strengths of both linguistic and statistical approaches to NLP can be combined in a single framework. The rules are contextual constraints for resolving syntactic ambiguities expressed as alternative tags, and the statistical language model consists of corpus-based n-grams of syntactic tags. The success of the hybrid syntactic disambiguator is evaluated against a held-out benchmark corpus. Also the contributions of the linguistic and statistical language models to the hybrid model are estimated.

1 Introduction

The language models used by natural language analyzers are traditionally based on two approaches. In the linguistic approach, the model is based on hand-crafted rules derived from the linguist's general and/or corpus-based knowledge about the object language. In the data-driven approach, the model is automatically generated from annotated text corpora, and the model can be represented e.g. as n-grams (Garside et al., 1987), local rules (Hindle, 1989) or neural nets (Schmid, 1994).

Most hybrid approaches combine statistical information with automatically extracted rule-based information (Brill, 1995; Daelemans et al., 1996). Relatively little attention has been paid to models where the statistical approach is combined with a truly linguistic model (i.e. one generated by a linguist). This paper reports one such approach: syntactic rules written by a linguist are combined with statistical information using the relaxation labelling algorithm.

Our application is very shallow parsing: identification of verbs, premodifiers, nominal and adverbial heads, and certain kinds of postmodifiers. We call this parser a noun phrase parser.

The input is English text morphologically tagged with a rule-based tagger called EngCG (Voutilainen et al., 1992; Karlsson et al., 1995). Syntactic wordtags are added as alternatives (e.g. each adjective gets a premodifier tag, postmodifier tag and a nominal head tag as alternatives). The system should remove contextually illegitimate tags and leave intact each word's most appropriate tag. In other words, the syntactic language model is applied by a disambiguator.

The parser has a *recall* of 100% if all words retain the correct morphological and syntactic reading; the system's *precision* is 100% if the output contains no illegitimate morphological or syntactic readings. In practice, some correct readings are discarded, and some ambiguities remain unresolved (i.e. some words retain two or more alternative analyses).

The system can use linguistic rules and corpus-based statistics. Notable about the system is that minimal human effort was needed for creating its language models (the linguistic consisting of syntactic disambiguation rules based on the Constraint Grammar framework (Karlsson, 1990; Karlsson et al., 1995); the corpus-based consisting of bigrams and trigrams):

- Only one day was spent on writing the 107 syntactic disambiguation rules used by the linguistic parser.

- No human annotators were needed for annotating the training corpus (218,000 words of journalese) used by the data-driven learning modules of this system: the training corpus was annotated by (i) tagging it with the EngCG morphological tagger, (ii) making the tagged text

syntactically ambiguous by adding the alternative syntactic tags to the words, and (iii) resolving most of these syntactic ambiguities by applying the parser with the 107 disambiguation rules.

The system was tested against a fresh sample of five texts (6,500 words). The system's recall and precision was measured by comparing its output to a manually disambiguated version of the text. To increase the objectivity of the evaluation, system outputs and the benchmark corpus are made publicly accessible (see Section 6).

Also the relative contributions of the linguistic and statistical components are evaluated. The linguistic rules seldom discard the correct tag, i.e. they have a very high recall, but their problem is remaining ambiguity. The problems of the statistical components are the opposite: their recall is considerably lower, but more (if not all) ambiguities are resolved. When these components are used in a balanced way, the system's overall recall is 97.2% – that is, 97.2% of all words get the correct analysis – and its precision is 96.1% – that is, of the readings returned by the system, 96.1% are correct.

The system architecture is presented in Figure 1.

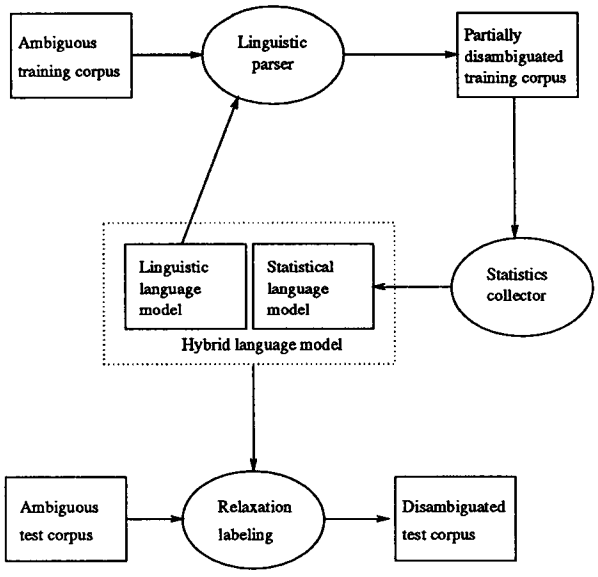

Figure 1: Parser architecture.

The structure of the paper is the following. First, we describe our general framework, the relaxation labelling algorithm. Then we proceed to the application by outlining the grammatical representation used in our shallow syntax. After this, the disambiguation rules and their development are described. Next in turn is a description of how the data-driven language model was generated. The evaluation of the system is then presented: first the preparation of the benchmark corpus is described, then the results of the tests are given. The paper ends with some concluding remarks.

2 The Relaxation Labelling Algorithm

Since we are dealing with a set of constraints and want to find a solution which optimally satisfies them all, we can use a standard Constraint Satisfaction algorithm to solve that problem.

Constraint Satisfaction Problems are naturally modelled as Consistent Labeling Problems (Larrosa and Meseguer, 1995). An algorithm that solves CLPs is Relaxation Labelling.

It has been applied to part-of-speech tagging (Padró, 1996) showing that it can yield as good results as a HMM tagger when using the same information. In addition, it can deal with any kind of constraints, thus the model can be improved by adding any other constraints available, either statistics, hand-written or automatically extracted (Màrquez and Rodríguez, 1995; Samuelsson et al., 1996).

Relaxation labelling is a generic name for a family of iterative algorithms which perform function optimisation, based on local information. See (Torras, 1989) for a summary.

Given a set of variables, a set of possible labels for each variable, and a set of compatibility constraints between those labels, the algorithm finds a combination of weights for the labels that maximises "global consistency" (see below).

Let $V = \{v_1, v_2, \ldots, v_n\}$ be a set of variables.
Let $t_i = \{t_1^i, t_2^i, \ldots, t_{m_i}^i\}$ be the set of possible labels for variable v_i.

Let CS be a set of constraints between the labels of the variables. Each constraint $C \in CS$ states a "compatibility value" C_r for a combination of pairs variable–label. Any number of variables may be involved in a constraint.

The aim of the algorithm is to find a weighted labelling[1] such that "global consistency" is maximised. Maximising "global consistency" is defined as maximising $\sum_j p_j^i \times S_{ij}$, $\forall v_i$, where p_j^i is the weight for label j in variable v_i and S_{ij} the support received by the same combination. The support for the pair variable–label expresses *how compatible* that pair is with the labels of neighbouring variables, according to the constraint set.

[1] A weighted labelling is a weight assignment for each label of each variable such that the weights for the labels of the same variable add up to one.

The support is defined as the sum of the influence of every constraint on a label.

$$S_{ij} = \sum_{r \in R_{ij}} \text{Inf}(r)$$

where:
R_{ij} is the set of constraints on label j for variable i, i.e. the constraints formed by any combination of variable–label pairs that includes the pair (v_i, t_j^i).
$\text{Inf}(r) = C_r \times p_{k_1}^{r_1}(m) \times \ldots \times p_{k_d}^{r_d}(m)$, is the product of the current weights[2] for the labels appearing in the constraint except (v_i, t_j^i) (representing *how applicable* the constraint is in the current context) multiplied by C_r which is the constraint compatibility value (stating *how compatible* the pair is with the context).

Briefly, what the algorithm does is:

1. Start with a random weight assignment.
2. Compute the support value for each label of each variable. (How compatible it is with the current weights for the labels of the other variables.)
3. Increase the weights of the labels more compatible with the context (support greater than 0) and decrease those of the less compatible labels (support less than 0)[3], using the updating function:

$$p_j^i(m+1) = \frac{p_j^i(m) \times (1 + S_{ij})}{\sum_{k=1}^{k_i} p_k^i(m) \times (1 + S_{ik})}$$

$$\text{where} \quad -1 \leq S_{ij} \leq +1$$

4. If a stopping/convergence criterion[4] is satisfied, stop, otherwise go to to step 2.

3 Grammatical representation

The input of our parser is morphologically analyzed and disambiguated text enriched with alternative syntactic tags, e.g.

```
"<others>"
   "other" PRON NOM PL @>N @NH
"<moved>"
   "move" <SV> <SVO> V PAST VFIN @V
"<away>"
   "away" ADV ADVL @>A @AH
"<from>"
   "from" PREP @DUMMY
"<traditional>"
   "traditional" A ABS @>N @N< @NH
"<jazz>"
   "jazz" <-Indef> N NOM SG @>N @NH
"<practice>"
   "practice" N NOM SG @>N @NH
   "practice" <SVO> V PRES -SG3 VFIN @V
```

Every indented line represents a morphological reading; the sample shows that some morphological ambiguities are not resolved by the rule-based morphological disambiguator, known as the EngCG tagger (Voutilainen et al., 1992; Karlsson et al., 1995).

Our syntactic tags start with the "@" sign. A word is syntactically ambiguous if it has more than one syntactic tags (e.g. *practice* above has three alternative syntactic tags). Syntactic tags are added to the morphological analysis with a simple lookup module. The syntactic parser's main task is disambiguating (rather than adding new information to the input sentence): contextually illegitimate alternatives should be discarded, while legitimate tags should be retained (note that also morphological ambiguities may be resolved as a side effect).

Next we describe the syntactic tags:

- @>N represents premodifiers and determiners.
- @N< represents a restricted range of postmodifiers and the determiner "enough" following its nominal head.
- @NH represents nominal heads (nouns, adjectives, pronouns, numerals, ING-forms and non-finite ED-forms).
- @>A represents those adverbs that premodify (intensify) adjectives (including adjectival ING-forms and non-finite ED-forms), adverbs and various kinds of quantifiers (certain determiners, pronouns and numerals).
- @AH represents adverbs that function as head of an adverbial phrase.
- @A< represents the postmodifying adverb "enough".
- @V represents verbs and auxiliaries (incl. the infinitive marker "to").

[2] $p_k^r(m)$ is the weight assigned to label k for variable r at time m.

[3] Negative values for support indicate *incompatibility*.

[4] The usual criterion is to stop when there are no more changes, although more sophisticated heuristic procedures are also used to stop relaxation processes (Eklundh and Rosenfeld, 1978; Richards et al. , 1981).

- @>CC represents words introducing a coordination ("either", "neither", "both").
- @CC represents coordinating conjunctions.
- @CS represents subordinating conjunctions.
- @DUMMY represents all prepositions, i.e. the parser does not address the attachment of prepositional phrases.

4 Syntactic rules

4.1 Rule formalism

The rules follow the Constraint Grammar formalism, and they were applied using the recent parser-compiler CG-2 (Tapanainen, 1996). The parser reads a sentence at a time and discards those ambiguity-forming readings that are disallowed by a constraint.

Next we describe some basic features of the rule formalism. The rule

```
REMOVE (@>N)
 (*1C <<< OR (@V) OR (@CS) BARRIER (@NH));
```

removes the premodifier tag @>N from an ambiguous reading if somewhere to the right (*1) there is an unambiguous (C) occurrence of a member of the set <<< (sentence boundary symbols) or the verb tag @V or the subordinating conjunction tag @CS, and there are no intervening tags for nominal heads (@NH).

This is a partial rule about coordination:

```
REMOVE (@>N)
 (NOT 0 (DET) OR (NUM) OR (A))
 (1C (CC))
 (2C (DET)) ;
```

It removes the premodifier tag if all three context-conditions are satisfied:

- the word to be disambiguated (0) is not a determiner, numeral or adjective,
- the first word to the right (1) is an unambiguous coordinating conjunction, and
- the second word to the right is an unambiguous determiner.

In addition to REMOVing, also SELECTing a reading is possible: when all context-conditions are satisfied, all readings but the one the rule was expressly about are discarded.

The rules can refer to words and tags directly or by means of predefined sets. They can refer not only to any fixed context positions; also reference to contextual patterns is possible. The rules never discard a last reading, so every word retains at least one analysis. On the other hand, an ambiguity remains unresolved if there are no rules for that particular type of ambiguity.

4.2 Grammar development

A day was spent on writing 107 constraints; about 15,000 words of the parser's output were proofread during the process. The routine was the following:

1. The current grammar (containing e.g. 2 rules) is applied to the ambiguous input in a 'trace' mode in which the parser also indicates, which rule discarded which analysis,
2. The grammarian observes remaining ambiguities and proposes new rules for disambiguating them, and
3. He also tries to identify misanalyses (cases where the correct tag is discarded) and, using the trace information, corrects the faulty rule

This routine is useful if the development time is very restricted, and only the most common ambiguity types have to be resolved with reasonable success. However, if the grammar should be of a very high quality (extremely few mispredictions, high degree of ambiguity resolution), a large test corpus, formally similar to the input except for the manually added extra information about the correct analysis, should be used. This kind of test corpus would enable the automatic identification of mispredictions as well as counting of various performance statistics for the rules. However, manually disambiguating a test corpus of a few hundred thousand words would probably require a human effort of at least a month.

4.3 Sample output

The following is genuine output of the linguistic (CG-2) parser using the 107 syntactic disambiguation rules. The traces starting with "S:" indicate the line on which the applied rule is in the grammar file. One syntactic (and morphological) ambiguity remains unresolved: *until* remains ambiguous due to preposition and subordinating conjunction readings.

```
"<aachen>" S:46
   "aachen" <*> <Proper> N NOM SG @NH
"<remained>"
   "remain" <SVC/N> <SVC/A> V PAST VFIN @V
"<a>"
   "a" <Indef> DET CENTRAL ART SG @>N
"<free>" S:316, 49
```

```
   "free" A ABS @>N
"<imperial>" S:49, 57
   "imperial" A ABS @>N
"<city>" S:46
   "city" N NOM SG @NH
"<until>"
   "until" PREP @DUMMY
   "until" <**CLB> CS @CS
"<occupied>" S:116, 345, 46
   "occupy" <SVO> PCP2 @V
"<by>"
   "by" PREP @DUMMY
"<france>" S:46
   "france" <*> <Proper> N NOM SG @NH
"<in>"
   "in" PREP @DUMMY
"<1794>" S:121, 49
   "1794" <1900> NUM CARD @NH
"<$.>"
```

5 Hybrid language model

To solve shallow parsing with the relaxation labelling algorithm we model each word in the sentence as a variable, and each of its possible readings as a label for that variable. We start with a uniform weight distribution.

We will use the algorithm to select the right syntactic tag for every word. Each iteration will increase the weight for the tag which is currently most compatible with the context and decrease the weights for the others.

Since constraints are used to decide *how compatible* a tag is with its context, they have to assess the compatibility of a combination of readings. We adapt CG constraints described above.

The **REMOVE** constraints express total incompatibility[5] and **SELECT** constraints express total compatibility (actually, they express incompatibility of all other possibilities).

The compatibility value for these should be at least as strong as the strongest value for a statistically obtained constraint (see below). This produces a value of about ± 10.

But because we want the linguistic part of the model to be more important than the statistical part and because a given label will receive the influence of about two bigrams and three trigrams[6], a single linguistic constraint might have to override five statistical constraints. So we will make the compatibility values six times stronger, that is, ± 60.

Since in our implementation of the CG parser (Tapanainen, 1996) constraints tend to be applied in a certain order – e.g. **SELECT** constraints are usually applied *before* **REMOVE** constraints – we adjust the compatibility values to get a similar effect: if the value for **SELECT** constraints is $+60$, the value for **REMOVE** constraints will be lower in absolute value, (i.e. -50). With this we ensure that two contradictory constraints (if there are any) do not cancel each other. The **SELECT** constraint will win, as if it had been applied before.

This enables using any Constraint Grammar with this algorithm although we are applying it more flexibly: we do not decide whether a constraint is applied or not. It is always applied with an influence (perhaps zero) that depends on the weights of the labels.

If the algorithm should apply the constraints in a more strict way, we can introduce an influence threshold under which a constraint does not have enough influence, i.e. is not applied.

We can add more information to our model in the form of statistically derived constraints. Here we use bigrams and trigrams as constraints.

The 218,000-word corpus of journalese from which these constraints were extracted was analysed using the following modules:

- EngCG morphological tagger
- Module for introducing syntactic ambiguities
- The NP disambiguator using the 107 rules written in a day

No human effort was spent on creating this training corpus. The training corpus is partly ambiguous, so the bi/trigram information acquired will be slightly noisy, but accurate enough to provide an *almost* supervised statistical model.

For instance, the following constraints have been statistically extracted from bi/trigram occurrences in the training corpus.

```
-0.415371 (@V)
        (1 (@>N));
```

[5] We model compatibility values using mutual information (Cover and Thomas, 1991), which enables us to use negative numbers to state *incompatibility*. See (Padró, 1996) for a performance comparison between M.I. and other measures when applying relaxation labelling to NLP.

[6] The algorithm tends to select one label per variable, so there is always a bi/trigram which is applied more significantly than the others.

```
4.28089 (@>A)
        (-1 (@>A))
        (1 (@AH));
```

The compatibility value is the mutual information, computed from the probabilities estimated from a training corpus. We do not need to assign the compatibility values here, since we can estimate them from the corpus.

The compatibility values assigned to the hand-written constraints express the strength of these constraints compared to the statistical ones. Modifying those values means changing the relative weights of the linguistic and statistical parts of the model.

6 Preparation of the benchmark corpus

For evaluating the systems, five roughly equal-sized benchmark corpora not used in the development of our parsers and taggers were prepared. The texts, totaling 6,500 words, were copied from the Gutenberg e-text archive, and they represent present-day American English. One text is from an article about AIDS; another concerns brainwashing techniques; the third describes guerilla warfare tactics; the fourth addresses the assassination of J. F. Kennedy; the last is an extract from a speech by Noam Chomsky.

The texts were first analysed by a recent version of the morphological analyser and rule-based disambiguator EngCG, then the syntactic ambiguities were added with a simple lookup module. The ambiguous text was then manually disambiguated. The disambiguated texts were also proofread afterwards. Usually, this practice resulted in one analysis per word. However, there were two types of exception:

1. The input did not contain the desired alternative (due to a morphological disambiguation error). In these cases, no reading was marked as correct. Two such words were found in the corpora; they detract from the performance figures.

2. The input contained more than one analyses all of which seemed equally legitimate, even when semantic and textual criteria were consulted. In these cases, all the equal alternatives were marked as correct. The benchmark corpus contains 18 words (mainly ING-forms and nonfinite ED-forms) with two correct syntactic analyses.

The number of multiple analyses could probably be made even smaller by specifying the grammatical representation (usage principles of the syntactic tags) in more detail, in particular incorporating some analysis conventions for certain apparent borderline cases (for a discussion of specifying a parser's linguistic task, see (Voutilainen and Järvinen, 1995)).

To improve the objectivity of the evaluation, the benchmark corpus (as well as parser outputs) have been made available from the following URLs:
http://www.ling.helsinki.fi/~avoutila/anlp97.html
http://www-lsi.upc.es/~lluisp/anlp97.html

7 Experiments and results

We tested linguistic, statistical and hybrid language models, using the CG-2 parser (Tapanainen, 1996) and the relaxation labelling algorithm described in Section 2.

The statistical models were obtained from a training corpus of 218,000 words of journalese, syntactically annotated using the linguistic parser (see above).

Although the linguistic CG-2 parser does not disambiguate completely, it seems to have an almost perfect recall (cf. Table 1 below), and the noise introduced by the remaining ambiguity is assumed to be sufficiently lower than the signal, following the idea used in (Yarowsky, 1992).

The collected statistics were bigram and trigram occurrences.

The algorithms and models were tested against a hand-disambiguated benchmark corpus of over 6,500 words.

We measure the performance of the different models in terms of recall and precision. Recall is the percentage of words that get the correct tag among the tags proposed by the system. Precision is the percentage of tags proposed by the system that are correct.

	CG-2 parser prec. - recall	**Rel. Labelling prec. - recall**
C	90.8% – 99.7%	93.3% – 98.4%

Table 1: Results obtained with the linguistic model.

	Rel. Labelling prec. - recall
B	87.4% – 88.0%
T	87.6% – 88.4%
BT	88.1% – 88.8%

Table 2: Results obtained with statistical models.

	Rel. Labelling prec. - recall
BC	96.0% – 97.0%
TC	95.9% – 97.0%
BTC	96.1% – 97.2%

Table 3: Results obtained with hybrid models.

Precision and recall results (computed on all words except punctuation marks, which are unambiguous) are given in tables 1, 2 and 3. Models are coded as follows: **B** stands for bigrams, **T** for trigrams and **C** for hand-written constraints. All combinations of information types are tested. Since the CG-2 parser handles only Constraint Grammars, we cannot test this algorithm with statistical models.

These results suggest the following conclusions:

- Using the same language model (107 rules), the relaxation algorithm disambiguates more than the CG-2 parser. This is due to the weighted rule application, and results in more misanalyses and less remaining ambiguity.
- The statistical models are clearly worse than the linguistic one. This could be due to the noise in the training corpus, but it is more likely caused by the difficulty of the task: we are dealing here with shallow syntactic parsing, which is probably more difficult to capture in a statistical model than e.g. POS tagging.
- The hybrid models produce less ambiguous results than the other models. The number of errors is much lower than was the case with the statistical models, and somewhat higher than was the case with the linguistic model. The gain in precision seems to be enough to compensate for the loss in recall[7].
- There does not seem to be much difference between BC and TC hybrid models. The reason is probably that the job is mainly done by the linguistic part of the model – which has a higher relative weight – and that the statistical part only helps to disambiguate cases where the linguistic model doesn't make a prediction. The BTC hybrid model is slightly better than the other two.
- The small difference between the hybrid models suggest that some reasonable statistics provide enough disambiguation, and that not very sophisticated information is needed.

[7]This obviously depends on the flexibility of one's requirements.

8 Discussion

In this paper we have presented a method for combining linguistic hand-crafted rules with statistical information, and we applied it to a shallow parsing task.

Results show that adding statistical information results in an increase in the disambiguation ratio, getting a higher precision. The price is a decrease in recall. Nevertheless, the risk can be controlled since more or less statistical information can be used depending on the precision/recall tradeoff one wants to achieve.

We also used this technique to build a shallow parser with minimal human effort:

- 107 disambiguation rules were written in a day.
- These rules were used to analyze a training corpus, with a very high recall and a reasonable precision.
- This slightly ambiguous training corpus is used for collecting bigram and trigram occurrences. The noise introduced by the remaining ambiguity is assumed not to distort the resulting statistics too much.
- The hand-written constraints and the statistics are combined using a relaxation algorithm to analyze the test corpus, rising the precision to 96.1% and lowering the recall only to 97.2%.

Finally, a reservation must be made: what we have not investigated in this paper is how much of the extra work done with the statistical module could have been done equally well or even better by spending e.g. another day writing a further collection of heuristic rules. As suggested e.g. by Tapanainen and Voutilainen (1994) and Chanod and Tapanainen (1995), hand-coded heuristics may be a worthwhile addition to 'strictly' grammar-based rules.

Acknowledgements

We wish to thank Timo Järvinen, Pasi Tapanainen and two ANLP'97 referees for useful comments on earlier versions of this paper.

The first author benefited from the collaboration of Juha Heikkilä in the development of the linguistic description used by the EngCG morphological tagger; the two-level compiler for morphological analysis in EngCG was written by Kimmo Koskenniemi; the recent version of the Constraint Grammar parser (CG-2) was written by Pasi Tapanainen. The Constraint Grammar framework was originally proposed by Fred Karlsson.

References

E. Brill. 1995. Unsupervised Learning of Disambiguation Rules for Part-of-speech Tagging. In *Proceedings of 3rd Workshop on Very Large Corpora*, Massachusetts.

J.-P. Chanod and P. Tapanainen 1995. Tagging French: comparing a statistical and a constraint-based method. In *Proc. EACL'95*. ACL, Dublin.

T.M. Cover and J.A. Thomas (Editors) 1991. *Elements of information theory.* John Wiley & Sons.

J. Eklundh and A. Rosenfeld. 1978. Convergence Properties of Relaxation Labelling. Technical Report no. 701. Computer Science Center. University of Maryland.

W. Daelemans, J. Zavrel, P. Berck and S. Gillis. 1996. MTB: A Memory-Based Part-of-Speech Tagger Generator. In *Proceedings of 4th Workshop on Very Large Corpora.* Copenhagen, Denmark.

R. Garside, G. Leech and G. Sampson (Editors) 1987. *The Computational Analysis of English.* London and New York: Longman.

D. Hindle. 1989. Acquiring disambiguation rules from text. In *Proc. ACL'89.*

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

F. Karlsson, A. Voutilainen, J. Heikkilä and A. Anttila. (Editors) 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text.* Mouton de Gruyter, Berlin and New York.

J. Larrosa and P. Meseguer. 1995. An Optimization-based Heuristic for Maximal Constraint Satisfaction. In *Proceedings of International Conference on Principles and Practice of Constraint Programming.*

L. Màrquez and H. Rodríguez. 1995. Towards Learning a Constraint Grammar from Annotated Corpora Using Decision Trees. ESPRIT BRA-7315 Acquilex II, Working Paper.

L. Padró. 1996. POS Tagging Using Relaxation Labelling. In *Proceedings of 16th International Conference on Computational Linguistics*, Copenhagen, Denmark.

J. Richards, D. Landgrebe and P. Swain. 1981. On the accuracy of pixel relaxation labelling. In *IEEE Transactions on System, Man and Cybernetics.* Vol. SMC-11

C. Samuelsson, P. Tapanainen and A. Voutilainen. 1996. Inducing Constraint Grammars. In *Proceedings of the 3rd International Colloquium on Grammatical Inference.*

H. Schmid 1994. Part-of-speech tagging with neural networks. In *Proceedings of 15th International Conference on Computational Linguistics*, Kyoto, Japan.

P. Tapanainen 1996. *The Constraint Grammar Parser CG-2.* Department of General Linguistics, University of Helsinki.

P. Tapanainen and A. Voutilainen 1994. Tagging accurately – Don't guess if you know. In *Proceedings of the 4th Conference on Applied Natural Language Processing, ACL.* Stuttgart.

C. Torras. 1989. Relaxation and Neural Learning: Points of Convergence and Divergence. *Journal of Parallel and Distributed Computing*, 6:217-244

A. Voutilainen, J. Heikkilä and A. Anttila 1992. *Constraint Grammar of English. A Performance-Oriented Introduction.* Publications 21, Department of General Linguistics, University of Helsinki.

A. Voutilainen and T. Järvinen. 1995. Specifying a shallow grammatical representation for parsing purposes. In *Proceedings of the 7th meeting of the European Association for Computational Linguistics.* 210-214.

D. Yarowsky. 1992. Word-sense disambiguations using statistical models of Roget's categories trained on large corpora. In *Proceedings of 14th International Conference on Computational Linguistics.* Nantes, France.