A Novel Framework for Reductionistic Statistical Parsing

Christer Samuelsson
Universität des Saarlandes
FR 8.7, Computerlinguistik, Postfach 1150
D-66041 Saarbrücken, Germany
Internet: `christer@coli.uni-sb.de`

Abstract

A reductionistic statistical framework for part-of-speech tagging and surface syntactic parsing is presented that has the same expressive power as the highly successful Constraint Grammar approach, see [Karlsson *et al* 1995]. The structure of the Constraint Grammar rules allows them to be viewed as conditional probabilities that can be used to update the lexical tag probabilities, after which low-probability tags are repeatedly removed.

Experiments using strictly conventional information sources on the Susanne and Teleman corpora indicate that the system performs as well as a traditional HMM-based part-of-speech tagger, yielding state-of-the-art results. The scheme also enables using the same information sources as the Constraint Grammar approach, and the hope is that it can improve on the performance of both statistical taggers and surface-syntactic analyzers.

1 Introduction

Part-of-speech (PoS) tagging consists in assigning to each word of an input text a tag from a finite set of possible tags, a tagset. The reason that this is a research issue is that a word can in general be assigned different tags depending on context. This assignment can be done in a number of different ways. One of these is statistical tagging, which is advocated in [Church 1988], [Cutting *et al* 1992] and many other articles. Here, the relevant information is extracted from large sets of often hand-tagged training data and fitted into a statistical language model, which is then used to assign the most likely tag to each word in the input text.

An alternative approach is taken in rule-based tagging, where linguistic knowledge is coded into rules that are applied to the input text to determine an appropriate tag for each word. The by far most successful work in this field is done in the Constraint Grammar [Karlsson *et al* 1995] and Finite-State Grammar [Koskenniemi *et al* 1992] frameworks. We will in this article focus on the former. First, however, we will describe what seems to be the standard procedure in statistical part-of-speech tagging, and then examine the Constraint Grammar approach a bit closer, before we discuss ways of combining them.

2 Traditional Statistical Part-of-Speech Tagging

This section describes a generic, but somewhat vanilla-flavoured statistical part-of-speech tagger. The tagger will use the following two information sources:

- **Lexical probabilities:**
  The probability of each tag T^i conditional on the word W that is to be tagged, $P(T^i \mid W)$.
- **Tag n-grams:**
  The probability of tag T^i at position k in the input string, denoted T^i_k, given that tags $T_{k-n+1} \dots T_{k-1}$ have been assigned to the previous $n-1$ words. Often n is set to two or three, and thus bigrams or trigrams are employed. In the latter case, this quantity is $P(T^i_k \mid T_{k-2}, T_{k-1})$.

A tagger is usually trained on a pretagged training corpus, which is divided into a training set, used to estimate these statistical parameters, and a set of held back data used to cope with sparse data by way of back-off smoothing. For example, the tag trigram probabilities might be estimated as follows:

$$P(T_k^i \mid T_{k-2}, T_{k-1}) \approx \lambda_3 f(T_k^i \mid T_{k-2}, T_{k-1}) + \lambda_2 f(T_k^i \mid T_{k-1}) + \lambda_1 f(T_k^i)$$

Here f is the relative frequence in the training set. The parameters $\lambda_j = \lambda_j(T_{k-2}, T_{k-1}, T_k)$ may depend on the particular tags, but are required to be nonnegative and to sum to one over j. Appropriate values for these parameters can be estimated using the held-out portion of the training corpus by employing any of a number of techniques; much used today are deleted interpolation [Brown *et al* 1992] and modified Good-Turing smoothing, [Gale & Church 1990].

Information sources $S_1, \ldots, S_n$ are combined by multiplying the scaled probabilities:

$$\frac{P(T \mid S_1, \ldots, S_n)}{P(T)} \approx \prod_{i=1}^{n} \frac{P(T \mid S_i)}{P(T)}$$

This formula can be established by Bayesian inversion, then performing the independence assumptions, and renewed Bayesian inversion:

$$P(T \mid S_1, \ldots, S_n) = \frac{P(T) \cdot P(S_1, \ldots, S_n \mid T)}{P(S_1, \ldots, S_n)} \approx$$

$$\approx P(T) \cdot \prod_{i=1}^{n} \frac{P(S_i \mid T)}{P(S_i)} = P(T) \cdot \prod_{i=1}^{n} \frac{P(T) \cdot P(S_i \mid T)}{P(T) \cdot P(S_i)} = P(T) \cdot \prod_{i=1}^{n} \frac{P(T \mid S_i)}{P(T)}$$

At runtime, a tagger typically works as follows: First, each word is assigned the set of all possible tags according to the lexicon. This will create a lattice. A dynamic programming technique is then used to find the sequence of tags $T_1 \ldots T_n$ that maximizes

$$P(T_1 \ldots T_n \mid W_1 \ldots W_n) =$$

$$= \prod_{k=1}^{n} P(T_k \mid T_1 \ldots T_{k-1}, W_1 \ldots W_n) \approx \prod_{k=1}^{n} P(T_k \mid T_{k-2}, T_{k-1}, W_k) \approx$$

$$\approx \prod_{k=1}^{n} \frac{P(T_k \mid T_{k-2}, T_{k-1}) \cdot P(T_k \mid W_k)}{P(T_k)} = \prod_{k=1}^{n} \frac{P(T_k \mid T_{k-2}, T_{k-1}) \cdot P(W_k \mid T_k)}{P(W_k)}$$

Since $P(W_k)$ does not depend on the tag sequence, we might as well maximize

$$\prod_{k=1}^{n} P(T_k \mid T_{k-2}, T_{k-1}) \cdot P(W_k \mid T_k)$$

The dynamic search algorithm (Viterbi search) employed to this end is well-described in for example [DeRose 1988].

HMM-based taggers work in exactly the same way; the abstract way in which they are viewed is however a bit different. In HMM-based tagging, $P(T_k \mid T_{k-2}, T_{k-1})$ is often referred to as the *language model* and $P(W_k \mid T_k)$ as the *signal model*. This is since each word is viewed as a signal emitted from some (hidden) internal state consisting of a tag in the bigram case, or a pair of tags in the trigram case. The signal model describes the probability of each word being emitted by some state (tag / tag pair) and the language model the probability of state (tag / tag pair) transitions. The tagging task then consists in finding the most likely sequence of states given the observed sequence of words. It is also possible to estimate the statistical parameters from untagged text using a lexicon and an initial bias, see [Rabiner 1989] for an excellent tutorial on HMMs as applied to speech and language processing in general.

3 The Constraint Grammar Approach

Although not yet fully realized, the basic philosophy behind the novel statistical approach proposed in the current article is to give it the same expressive power as the highly successful Constraint Grammar system. This system performs remarkably well; [Voutilainen & Heikkilä 1994] report 99.7 percent recall, or 0.3 percent error rate, which is ten times smaller than that of the best statistical taggers. These impressive results are achieved by:

1. Utilizing a number of different information sources, and not only the stereotyped lexical statistics and n-gram tag statistics that have become the de facto standard in statistical part-of-speech tagging. For example, it uses linguistically motivated rules referring to an arbitrarily long context.

2. Not fully resolving all ambiguities when this would jeopardize the recall. In fact, the quoted 99.7 percent recall corresponds to a remaining ambiguity of approximately 1.05 tags per word, or a precision of 95 percent.

3. Evaluating on more consistently annotated data than normally employed in the field.

4. Employing an appropriate tag set, amongst other things avoiding to introduce ambiguities based on subtle distinctions of relatively small importance.

Strategy 2 means that the system trades precision for recall, making it ideal as a preprocessor for natural language systems performing deeper analysis. It has been argued that this is the sole source of the success of the Constraint Grammar approach. The experiments in this article and [de Marcken 1990] clearly indicate otherwise. The corresponding tradeoff points for statistical PoS tagging employing traditional information sources are, in the best cases, 99.2% recall at 2.17 tags per word and 97.2% recall at 1.05 tags per word, see Table 2. This is lightyears away from the 99.7% recall at 1.05 tags per word achieved by the Constraint Grammar system.

Strategy 3 is controversial as it is sometimes claimed that expert human evaluators will invariably disagree on approximately three percent of the tags, and that thus the reported error rate of 0.3 percent is theoretically impossible. We argue that: a) This is not the case. b) Even if it were the case, it is irrelevant. c) Finally, even if it were in fact relevant, it would still be consistent with the reported results. In addition to this, other experiments indicate that it is in fact possible to achieve 100 percent inter-judge agreement, see [Voutilainen & Järvinen 1995].

Firstly, this claim is based on the fact that there is a three percent disagreement amongst the persons who annotated the Brown corpus [Francis & Kucera 1982]. This does not constitute solid empirical evidence that human annotators are incapable of much more consistent hand annotation — It merely states that on one occasion, a few linguists disagreed in three percent of the cases. Anyone who has ever tried to get even a small number of linguists to agree on a choice of restaurant, not to mention on a treatment of some linguistic phenomenon, will find this figure surprisingly low. This problem seems to be more social than scientific.

In fact, if this were actually true, it would introduce some very strange effects. For example, every twelve-word sentence would have a 30 percent chance of having at least one word tagged differently by any two observers; every 23-word sentence would have a 50-percent chance of this. If one observer is the author or utterer of the sentence, and the other is a reader or a listener, and it is crucial for understanding that the receiver gets the tags right, this would render human-human communication virtually impossible. If the exact tags do not really matter, why bother with PoS tagging in the first place? Or why not, like the Constraint Grammar system, avoid trying to disambiguate these apparently irrelevant ambiguities?

Secondly, even if there were an upper limit to the level of agreement amongst any set of human classifiers, this would not necessarily imply that there is no automatic procedure for performing the classification task with much greater accuracy. Assume that a number of professors of mathematics were given limited time to individually and manually classify the numbers between 1,000,000 and 1,100,000 into primes and non-primes. No one would seriously argue that it is impossible to automatically determine whether or not a number is a prime based on the highly likely outcome that all the professors did not arrive at identical classifications. In fact, mathematicians would most likely manually carry out an automatic procedure in the nonobvious cases, given a sufficient amount of time.

Thirdly, in the quoted experiments, the Constraint Grammar system left in five percent ambiguity, which could easily accommodate the cases where the human evaluators are claimed to necessarily disagree.

More importantly, the Constraint Grammar system is being used by an increasing number of people for robust surface-syntactic analysis. Although the syntactic analysis is shallow, and by no means as accurate as the morphological analysis, it has for example still proved sufficient for the system to constitute the sole language-analysis component of a speech interface to a virtual-reality environment, see [Karlgren *et al* 1995].

The Constraint Grammar system works as follows: First, the input string is assigned all possible tags from the lexicon, or rather, from the morphological analyzer. Then, tags are removed iteratively by repeatedly applying a set of rules, or constraints, to the tagged string. The rules are applied one by one, and the order in which they are applied is not important. Thus the effects of the various information sources are separated. When no more tags are removed by the last iteration, the process terminates, and morphological disambiguation is concluded. Then a set of syntactic tags are assigned to the tagged input string and a similar process is performed for syntactic disambiguation. This method is often referred to as *reductionistic tagging*.

The rules are formulated as finite state automata, which allows very fast processing. In more detail, the rules could be formulated as finite-state transducers, but they are not. Instead, an equivalent, and very efficient intermediate format between the original rule format and finite-state transducers is employed [Tapanainen, personal communication].

Each rule applies to a current word with a set of candidate tags. The structure of a rule is typically:

> "In this and this context, discard the following tags."

or

> "In this and this context, commit to the following tag."

We will call discarding or committing to tags the *rule action*. A typical *rule context* is:

> "There is a word to the left that is unambiguously tagged with the following tag, and there are no intervening words tagged with such and such tags."

These rules are hand-coded by a skilled linguist, a laborious and time-consuming task. (Although [Chanod & Tapanainen 1995] indicates differently.)

4 A Novel Reductionistic Statistical Tagger

The structure of the Constraint Grammar rules readily allows their contexts to be viewed as the conditionings of conditional probabilities, and the actions have an obvious interpretation as the corresponding probabilities.

Each context type can be seen as a separate information source, and we will again combine information sources $S_1, \ldots, S_n$ by multiplying the scaled probabilities:

$$\frac{P(T \mid S_1, \ldots, S_n)}{P(T)} \approx \prod_{i=1}^{n} \frac{P(T \mid S_i)}{P(T)} \tag{1}$$

The context will in general not be fully disambiguated. Rather than employing dynamic programming over the lattice of remaining candidate tags, the new approach uses the weighted average over the remaining candidate tags to estimate the probabilities:

$$P(T \mid \cup_{i=1}^{n} C_i) = \sum_{i=1}^{n} P(T \mid C_i) \cdot P(C_i \mid \cup_{i=1}^{n} C_i) \tag{2}$$

It is assumed that $\{C_i : i = 1,\ldots,n\}$ constitutes a partition of the context C, i.e., that $C = \cup_{i=1}^{n} C_i$ and that $C_i \cap C_j = \emptyset$ for $i \neq j$. In particular, trigram probabilities are combined as follows:

$$P(T \mid C) = \sum_{(T_l,T_r)\in C} P(T \mid T_l,T_r) \cdot P((T_l,T_r) \mid C)$$

Here T denotes a candidate tag of the current word, T_l denotes a candidate tag of the immediate left neighbour, and T_r denotes a candidate tag of the immediate right neighbour. C is the set of ordered pairs (T_l,T_r) drawn from the set of candidate tags of the immediate neighbours. $P(T \mid T_l,T_r)$ is the symmetric trigram probability (centered around the current word).

The tagger works as follows: First tag probabilities are assigned to the input word string on purely lexical basis. Then the probabilities are updated using the various contextual information sources, corresponding to the rules of the Constraint Grammar, according to Eq. 1. Then low-probability candidate tags are removed and the probabilities are recalculated. The process terminates when the probabilities have stabilized and no more tags can be removed without jeopardizing the recall. The latter is accomplished by only removing candidate tags if their probabilities are below a certain threshold value.

As pointed out, there is no restriction on what information sources can be used; anything that can be formulated as a Constraint Grammar rule constitutes a valid conditional probability. It is of course a relevant question whether or not the independence assumptions behind Eq. 1 are valid; or, more importantly, if the resulting language model proves useful. The latter can only be evaluated empirically.

The most challenging task, which has yet only started, is to devise methods for automatically extracting "rules" from (pretagged) corpora. The current approach is to find conditionings that drastically alter the probability distributions compared to very similar conditionings; then the extra information must be decisive. Exploring the entire space of possible conditionings and comparing them is intractable, and heuristics are needed to guide the search. A simple rule that has been automatically extracted is based on the fact that the distribution is quite different when we know that there is a determiner to the left of the current word, but that there are no intervening nouns, as opposed to when there are intervening nouns.

Another thing that has not been investigated is how to incorporate the search for clause boundaries into the new framework. This could be done in a number of ways: One would be to insert them in a preprocessing step and treat them as facts in the further processing. Another approach would be to treat them the same way as other tags, and use Eq. 2 to handle uncertainty. Also this is an empirical question.

One would expect a slight decrease in performance using weighted averages instead of separate paths for handling ambiguous contexts. Incidentally, this is the main difference between the Constraint Grammar and Finite-State Grammar approaches; the rules of the latter can refer to paths through the lattice, something the rules of the former cannot. In order to ascertain that the proposed statistical method is not greatly inferior to traditional statistical PoS tagging when employing strictly conventional information sources, a series of experiments have been carried out as described in the following section.

5 Experiments

The novel reductionistic tagger was compared with a traditional HMM-based tagger; the latter is described in [Brants & Samuelsson 1995]. Both taggers employed strictly conventional information sources — lexical and trigram statistics. The experiments were carried out on two different corpora, the Susanne and Teleman corpora, using both the original and a reduced tagset. The Susanne corpus [Sampson 1995] is a re-annotated part of the Brown corpus [Francis & Kucera 1982] of contemporary English, and the Teleman corpus [Teleman 1974] is a corpus of contemporary Swedish, both comprising a variety of different text genres.

For the experiments, both corpora were divided into three sets, one large set (A) and two smaller sets (B and C). Three different divisions into training and test sets were used. First, all three sets were used for both training and testing. In the second and third case, training and test sets were disjoint, the large set and one of the small sets were used for training, the remaining small set was used for testing. To indicate what is gained by taking the context into account, an additional set of experiments using only lexical probabilities and ignoring context were performed as a baseline.

Unknown words were handled by creating a decision tree of the four last letters from words with three or less occurrences. Each node in the tree was associated with a probability distribution (over the tagset) extracted from these words, and the probabilities were smoothened through linear successive abstraction, see [Brants & Samuelsson 1995].

There were two cut-off values for contexts: Firstly, any context with less than 10 observations was discarded. Secondly, any context where the probability distributions did not differ substantially from the unconditional one was also discarded. Only the remaining ones were used for disambiguation. Due to the computational model employed, omitted contexts are equivalent to backing off to whatever the current probability distribution is. The distributions conditional on contexts are however susceptible to the problem of sparse data. This was handled using partial successive abstraction as described in [Brants & Samuelsson 1995].

The results are shown in Tables 1 and 2. [1] They clearly indicate that:

- Using contextual information, i.e., trigrams, improves tagging accuracy.
- The performance of the reductionistic tagger is on par with the HMM tagger and comparable to state-of-the-art statistical part-of-speech taggers.
- Tagging the Teleman corpus is the more difficult task.

The results using the Susanne corpus are similar to those reported for the Lancaster-Oslo-Bergen (LOB) corpus in [de Marcken 1990], where a statistical N-best-path approach was employed to trade precision for recall.

The tagging speed was typically a couple of hundred words per second on a SparcServer 1000, but varied with the size of the tagset and the amount of remaining ambiguity.

6 Conclusions

It is reassuring to see that the reductionistic tagger performs as well as the HMM tagger, indicating that the new framework is as powerful as the conventional one when using strictly conventional information sources. The new framework also enables using the same sort of information as the highly successful Constraint Grammar approach, and the hope is that the addition of further information sources can advance the performance of statistical taggers.

Viewed as an extension of the Constraint Grammar approach, the new scheme allows making decisions on the basis of not fully disambiguated portions of the input string. The absolute value of the probability of each tag can be used as a quantitative measure of when to remove a particular candidate tag and when to leave in the ambiguity. This provides a quantitative tool to control the tradeoff between recall (accuracy) and precision (remaining ambiguity).

Extracting data directly from corpora, rather than constructing rules by introspection, as is currently the case when developing constraint grammars, is less susceptible to human error, and should consequently result in less brittle systems. Thus the proposed method can most likely also constitute an improvement on surface-syntactic analyzers.

Acknowledgments

I would like to thank Thorsten Brants and Atro Voutilainen for comments and suggestions to improvements, and the anonymous reviewers for their constructive criticism of Section 3. I am most grateful to Thorsten Brants for providing the experimental data using the HMM tagger.

[1] It is not very interesting to compare the accuracy for the same threshold probability, but rather for the same remaining ambiguity.

Table 1: Results of the reductionistic experiments with the Teleman corpus

Training	Testing	Threshold:	0.00	0.05	0.075	0.10	0.15	0.20	0.30	0.50	HMM
Small Tagset											
A,B,C	A,B,C	Trigram and lexical statistics									
		Recall (%)	100.00	99.02	98.66	98.35	97.78	97.37	96.65	95.55	96.22
		Tags/word	2.38	1.15	1.12	1.10	1.07	1.05	1.03	1.00	1.00
		Lexical statistics only									
		Recall (%)	100.00	98.96	98.53	98.29	97.69	97.28	96.36	95.10	95.13
		Tags/word	2.38	1.25	1.17	1.14	1.09	1.07	1.03	1.00	1.00
A,B	C	Trigram and lexical statistics									
		Recall (%)	98.98	97.72	97.25	96.81	96.20	95.53	94.67	93.34	92.88
		Tags/word	2.54	1.21	1.17	1.14	1.10	1.07	1.04	1.00	1.00
		Lexical statistics only									
		Recall (%)	98.98	97.61	97.14	96.87	96.15	95.63	94.26	92.55	89.27
		Tags/word	2.54	1.34	1.25	1.21	1.14	1.11	1.04	1.00	1.00
A,C	B	Trigram and lexical statistics									
		Recall (%)	98.99	97.80	97.44	96.94	96.34	95.84	98.81	93.50	92.81
		Tags/word	2.51	1.23	1.18	1.15	1.11	1.08	1.04	1.00	1.00
		Lexical statistics only									
		Recall (%)	98.99	97.67	97.33	97.07	96.45	95.84	94.34	92.52	90.42
		Tags/word	2.51	1.34	1.26	1.21	1.14	1.10	1.04	1.00	1.00
Large Tagset											
A,B,C	A,B,C	Trigram and lexical statistics									
		Recall (%)	100.00	98.36	97.92	97.54	97.03	96.41	95.31	93.75	98.35
		Tags/word	3.69	1.23	1.18	1.15	1.11	1.08	1.04	1.00	1.00
		Lexical statistics only									
		Recall (%)	100.00	98.30	97.63	97.20	96.67	95.57	93.65	90.59	90.65
		Tags/word	3.69	1.43	1.31	1.26	1.22	1.16	1.08	1.00	1.00
A,B	C	Trigram and lexical statistics									
		Recall (%)	97.46	94.93	93.94	93.35	92.35	91.15	88.53	85.56	83.78
		Tags/word	4.16	1.47	1.37	1.31	1.24	1.18	1.08	1.00	1.00
		Lexical statistics only									
		Recall (%)	97.46	95.23	94.24	93.69	92.93	91.51	87.92	83.62	78.84
		Tags/word	4.16	1.69	1.53	1.44	1.34	1.26	1.11	1.00	1.00
A,C	B	Trigram and lexical statistics									
		Recall (%)	96.64	94.04	93.00	92.09	90.92	89.46	86.94	83.58	81.01
		Tags/word	4.18	1.48	1.38	1.32	1.24	1.18	1.08	1.00	1.00
		Lexical statistics only									
		Recall (%)	96.64	94.51	93.27	92.50	91.02	89.68	85.86	81.69	78.05
		Tags/word	4.18	1.71	1.54	1.44	1.34	1.24	1.10	1.00	1.00

Table 2: Results of the reductionistic experiments with the Susanne corpus

Training	Testing	Threshold:	0.00	0.05	0.075	0.10	0.15	0.20	0.30	0.50	HMM
Small Tagset											
A,B,C	A,B,C	Trigram and lexical statistics									
		Recall (%)	100.00	99.46	99.35	99.23	99.03	98.82	98.43	97.75	98.35
		Tags/word	2.07	1.08	1.07	1.06	1.04	1.03	1.02	1.00	1.00
		Lexical statistics only									
		Recall (%)	100.00	99.33	99.20	98.94	98.67	98.10	97.43	95.28	95.28
		Tags/word	2.07	1.18	1.16	1.14	1.11	1.08	1.05	1.00	1.00
A,B	C	Trigram and lexical statistics									
		Recall (%)	99.22	98.43	98.28	98.11	97.78	97.43	96.91	95.99	95.76
		Tags/word	2.23	1.14	1.11	1.09	1.07	1.05	1.02	1.00	1.00
		Lexical statistics only									
		Recall (%)	99.22	98.27	98.03	97.78	97.45	96.80	96.15	93.42	91.41
		Tags/word	2.23	1.25	1.23	1.19	1.15	1.11	1.08	1.00	1.00
A,C	B	Trigram and lexical statistics									
		Recall (%.)	99.22	98.46	98.22	97.99	97.58	97.15	96.49	95.54	95.18
		Tags/word	2.17	1.13	1.10	1.09	1.06	1.05	1.02	1.00	1.00
		Lexical statistics only									
		Recall (%)	99.22	98.21	97.88	97.61	97.35	96.47	95.46	92.87	91.20
		Tags/word	2.17	1.24	1.21	1.17	1.15	1.10	1.06	1.00	1.00
Large Tagset											
A,B,C	A,B,C	Trigram and lexical statistics									
		Recall (%)	100.00	99.25	99.12	98.96	98.74	98.44	98.04	96.87	99.80
		Tags/word	2.61	1.10	1.08	1.07	1.06	1.04	1.03	1.00	1.00
		Lexical statistics only									
		Recall (%)	100.00	99.05	98.88	98.59	98.20	97.58	96.72	93.98	93.98
		Tags/word	2.61	1.23	1.20	1.17	1.14	1.10	1.07	1.00	1.00
A,B	C	Trigram and lexical statistics									
		Recall (%)	98.31	96.94	96.52	96.19	95.68	95.02	94.21	92.70	92.61
		Tags/word	3.01	1.22	1.18	1.15	1.11	1.08	1.04	1.00	1.00
		Lexical statistics only									
		Recall (%)	98.31	96.91	96.49	95.94	95.50	94.40	93.42	90.26	86.98
		Tags/word	3.01	1.41	1.35	1.28	1.20	1.14	1.08	1.00	1.00
A,C	B	Trigram and lexical statistics									
		Recall (%)	98.49	97.03	96.72	96.41	95.88	95.16	94.29	92.71	93.07
		Tags/word	2.83	1.21	1.18	1.15	1.11	1.08	1.04	1.00	1.00
		Lexical statistics only									
		Recall (%)	98.49	96.95	96.55	96.05	95.57	94.44	93.26	90.31	88.16
		Tags/word	2.83	1.36	1.31	1.25	1.19	1.13	1.08	1.00	1.00

References

[Brants & Samuelsson 1995] Thorsten Brants and Christer Samuelsson. "Tagging the Teleman Corpus", in *Procs. 10th Nordic Conference on Computational Linguistics*, pp. 7–20, 1995.

[Brown *et al* 1992] P. F. Brown, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer and P. S. Roossin. "Class-based n-gram models of natural language", *Computational Linguistics 18(4)* pp. 467–479, 1992.

[Chanod & Tapanainen 1995] Jean-Pierre Chanod and Pasi Tapanainen. "Tagging French – Comparing a Statistical and a Constraint-Based Method", in *Procs. 7th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 149–156, ACL 1995.

[Church 1988] Kenneth W. Church. "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text", in *Procs. 2nd Conference on Applied Natural Language Processing*, pp. 136–143, 1988.

[Cutting *et al* 1992] Douglass R. Cutting, Julian Kupiec, Jan Pedersen and Penelope Sibun. "A Practical Part-of-Speech Tagger". in *Procs. 3rd Conference on Applied Natural Language Processing*, pp. 133–140, ACL, 1992.

[DeRose 1988] Steven J. DeRose. "Grammatical Category Disambiguation by Statistical Optimization", in *Computational Linguistics 14(1)*, pp. 31–39, 1988.

[Francis & Kucera 1982] N. W. Francis and H. Kucera. *Frequency Analysis of English Usage*, Houghton Mifflin, Boston, 1982.

[Gale & Church 1990] W. A. Gale and K. W. Church. "Poor Estimates of Context are Worse than None", in *Proc. of the Speech and Natural Language Workshop*, pp. 283–287, Morgan Kaufmann, 1990.

[Karlgren *et al* 1995] Jussi Karlgren, Ivan Bretan, Niklas Frost and Lars Jonsson. "Interaction Models, Reference, and Interactivity for Speech Interfaces to Virtual Environments", in *Procs. 2nd Eurographics Workshop on Virtual Environments — Realism and Real Time*, Monte Carlo 1995.

[Karlsson *et al* 1995] Fred Karlsson, Atro Voutilainen, Juha Heikkilä and Arto Anttila (eds). *Constraint Grammar. A Language-Independent System for Parsing Unrestricted Text*, Mouton de Gruyter, Berlin / New York, 1995.

[Koskenniemi *et al* 1992] Kimmo Koskenniemi, Pasi Tapanainen and Atro Voutilainen. "Compiling and Using Finite-State Syntactic Rules", in *Procs. 14th International Conference on Computational Linguistics*, pp. 156–162, ICCL 1992.

[de Marcken 1990] Carl G. de Marcken. "Parsing the LOB Corpus", in *Procs. 28th Annual Meeting of the Association for Computational Linguistics*, pp. 243–251, ACL 1990.

[Rabiner 1989] L. R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition", in *Proceedings of the IEEE 77(2)*, pp. 257–285, 1989.

[Sampson 1995] Geoffrey Sampson. *English for the Computer*, Oxford University Press, 1995.

[Teleman 1974] Ulf Teleman. *Manual för grammatisk beskrivning av talad och skriven svenska*, (in Swedish), Studentlitteratur, Lund, Sweden 1974.

[Voutilainen & Heikkilä 1994] Atro Voutilainen and Juha Heikkilä. "An English constraint grammar (ENGCG): a surface-syntactic parser of English", in *Procs. 14th International Conference on English Language Research on Computerized Corpora*, pp. 189–199, Zürich, 1994.

[Voutilainen & Järvinen 1995] Atro Voutilainen and Timo Järvinen. "Specifying a shallow grammatical representation for parsing purposes", in *Procs. 7th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 210–214, ACL 1995.