Extending the View
Explorations in Bootstrapping a Swedish PoS Tagger

Eva Forsbom
Department of Linguistics and Philology, Uppsala University
Graduate School of Language Technology
evafo@stp.lingfil.uu.se

Abstract

State-of-the-art statistical part-of-speech taggers mainly use information on tag bi- or trigrams, depending on the size of the training corpus. Some also use lexical emission probabilities above unigrams with beneficial results. In both cases, a wider context usually gives better accuracy for a large training corpus, which in turn gives better accuracy than a smaller one. Large corpora with validated tags, however, are scarce, so a bootstrap technique can be used. As the corpus grows, it is probable that a widened context would improve results even further.

In this paper, we looked at the contribution to accuracy of such an extended view for both tag transitions and lexical emissions, applied to both a validated Swedish source corpus and a raw bootstrap corpus. We found that the extended view was more important for tag transitions, in particular if applied to the bootstrap corpus. For lexical emission, it was also more important if applied to the bootstrap corpus than to the source corpus, although it was beneficial for both. The overall best tagger had an accuracy of 98.05%.

1 Introduction

Given the limitations of computational and human resources, state-of-the-art statistical taggers mostly use context information on tag bigrams, for smaller training corpora, or trigrams, for larger training corpora. Some also use lexical emission probabilities above unigrams, although with a rather limited context view, with beneficial results (e.g. Thede and Harper, 1999; Toutanova et al., 2003). But as computational power grows, and (semi)automatic annotation becomes more correct over time, resulting in large almost-correct training corpora, it would be interesting to see if it's worth extending the view.

For Swedish, several statistical part-of-speech taggers have been trained on the Swedish Stockholm-Umeå Corpus (SUC, Ejerhed et al., 2006), which has become a *de facto* standard for training and evaluating part-of-speech taggers. Most of them are based on hidden Markov models (e.g. Carlberger and Kann, 1999; Hall, 2003; Megyesi, 2002; Nivre, 2000; Sjöbergh, 2003b), with bi- or trigram tag transition probabilities.

As SUC is a balanced corpus (not just news texts) with a fairly large tagset, it is too small to be used alone as training data for any higher-accuracy tagger, so it has also been used to bootstrap a much larger, unannotated, corpus, that can be added as training data. In previous studies, bootstrapping has proved to be a viable approach (cf. Forsbom, 2008b; Merialdo, 1994; Nivre and Grönqvist, 2001; Sjöbergh, 2003a).

A recent open-source tagger, HunPos (Halácsy et al., 2007), include the range of parameters we would like to explore for extended context views for tag transition and lexical emissions.

In the following, we first describe the method, tagger and data sets used (Section 2), before describing the parameters used (Section 3). Results from experimental runs are then discussed and explored using a regression tree (Section 4).

2 Bootstrapping

In order to explore the effect of extending the view, large corpora are needed. Unfortunately, large validated training corpora are scarce, so in the abscence of such a desired resource, we have to build our

Kristiina Jokinen and Eckhard Bick (Eds.)
NODALIDA 2009 Conference Proceedings, pp. 34–40

own. And we do this by using a smaller-sized validated (source) corpus to bootstrap an order of magnitude larger (bootstrap) corpus, which will contain some noise, but in general, will be correct.

2.1 Method

The following bootstrap procedure was used:

1. Train a training model on the entire source corpus.
2. Tag the bootstrap corpus using the training model.
3. Train an evaluation model on the tagged bootstrap corpus (not including the source corpus). For other taggers than TnT (Brants, 2000), train a TnT lexical model on the same data, to use for evaluation statistics on known/unknown words.
4. Evaluate the evaluation model on 10 folds of the source corpus (if possible, drilled-down by genre).
5. (Train a final tag model on a concatenation of the source corpus and the tagged bootstrap corpus.)

The procedure is part of an ongoing project where various taggers and bootstrap corpora are compared (cf. Forsbom, 2006, 2008a,b). Therefore, evaluation is done with the same evaluation program, `tnt-diff`, to get comparable results on known/unknown words regardless of tagger. The known/unknown statistics should therefore be seen from a "TnT perspective", while the overall results are tagger-neutral.

Although we do not use proper 10-fold cross-validation (as we use the entire source corpus for bootstrapping), we still evaluate separately on 10 folds to be able to measure standard deviation.

In the optional fifth step, a final tag model which includes the source corpus and most likely gives even better results, could be trained and used in applications. Models from the experiment reported here are, for example, used in two other projects for summarisation and measuring readability.

2.2 Tagger

In this experiment, we use HunPos (Halácsy et al., 2007), which is a recent open-source implementation of many of the features included in TnT (Brants, 2000). As hidden Markov model taggers, both use a state transition probability for the current tag given a history of previous tags, and a lexical emission probability for the current word given a history of previous tags (see further in Section 3).

Unknown words are handled by suffix probability estimates from low-frequency words. HunPos also uses the same linear interpolation smoothing technique as in TnT. For HunPos, it is currently the only smoothing choice, while TnT also includes alternative techniques. If HunPos is trained using trigram state transitions and unigram lexical emission, it behaves as TnT with default settings.

The main reason for using HunPos here is the possibility to vary the history both for state transitions and lexical emissions, while in TnT, the history for lexical emission is fixed and for state transition limited to uni-, bi-, and trigrams.

2.3 Source corpus

We have chosen to use SUC (Ejerhed et al., 2006) as a source corpus for two reasons apart from it being a *de facto* standard: it contains validated tags, and it is a balanced corpus, and therefore possibly a better representative of general language than a single-genre corpus.

SUC contains modern Swedish prose covering approximately 1.2 million word tokens. The 1,040 text samples are from the years 1990 to 1994, and are meant to mirror what a Swedish person might read in the early nineties.

The distribution of tokens between genres (or main categories) is shown in Table 1.

ID	Genre	Tokens (%)
a	Press: Reportage	9.1
b	Press: Editorial	3.5
c	Press: Reviews	5.6
e	Skills and Hobbies	11.5
f	Popular Lore	9.4
g	Biographies, essays	5.2
h	Miscellaneous	13.9
j	Learned and scientific writing	16.4
k	Imaginative prose	25.4

Table 1: Distribution of tokens/genre in SUC.

2.3.1 Choice of tagset

The SUC corpus has two interchangeable tagsets: SUC (Ejerhed et al., 1992) and PAROLE (see Section 2.4). An alternative to the SUC tagset is the Granska tagset, which in general gives better accuracy (2% improvement). The Granska tagset is a slight modification of the SUC tagset. Modifications include merging of infrequent tags, adding

information on auxiliary verbs, reclassification of present participles to adjectives, and adding information on set- and date-describing words (Carlberger and Kann, 1999).

In a study of the contribution of the modifications, Forsbom (2008a) found that a distinction between main and auxiliary verbs was beneficial for copulas and temporal auxiliaries, but maybe not for modal verbs. The addition of the number feature singular to singular numbers, and the semantic feature date to names of days and months, was also beneficial, but to a lesser degree. These modifications are revertible without loss of information.

Some other modifications were also beneficial, but not revertible, e.g. conflation of past participle tags with the corresponding tags for adjectives.

To benefit from the improved accuracy that some of the Granska tags give, we have here used the SUC tagset with revertible Granska modifications for copulas, auxiliaries, singular numbers, and dates.

A comparison of accuracy for the three tagsets is shown in Table 2[1].

Tagset	Overall	Known	Unknown
SUC	95.52±0.15	96.31±0.13	86.26±0.99
Granska	95.68±0.14	96.42±0.13	87.09±0.91
Modified	95.61±0.14	96.40±0.12	86.37±0.96

Table 2: Estimated accuracy and standard deviation for the SUC, Granska and modified tagsets (10-fold cross-validation on SUC). Proportion of unknown words is 7.87 ± 0.20.

2.4 Bootstrap corpus

There are not many available large corpora of Swedish texts, and even fewer balanced corpora representing general language. Of the ones that do exist, the balanced Swedish PAROLE corpus has been used with success for bootstrapping (Forsbom, 2008b). The PAROLE corpus (University of Gothenburg) was collected for the EU project PAROLE (Preparatory Action for Linguistic Resources Organisation for Language Engineering) finished in 1997. The corpus contains around 19.4 million words of written texts from various categories, mainly sampled from The Swedish Language Bank (see Table 3). The texts have been part-of-speech tagged with PAROLE tags using a statistical tagger by Daniel Ridings (University of Gothenburg).

[1]The comparison was done with TnT.

Text category	Period	Tokens (%)
Novels	1976–1981	22.7
Newspapers	1976–1997	70.1
Magazines	1995–1996	2.1
Web texts	1997	5.2

Table 3: Distribution of tokens/genre in PAROLE.

In order to harmonise the PAROLE corpus with SUC, we made some changes to the original corpus:

- A set of known multi-word abbreviations have been treated as one token, with any whitespace replaced by an underscore.
- Sentence boundaries have been introduced with a simplistic sentence splitter (i.e. new sentence after .,!,? if the following line starts with capital, digit, or -).
- The original tags were replaced during bootstrap by the modified tagset used here.

3 Exploring possible views

We were interested in seeing the effect of widening the view, from the commonly used bi- or trigrams to as high an n-gram as we could compute. In the hidden Markov model, the state transition probability of a tag is based on the previous k tags (the tag order). For the default trigram tag order, $k = 2$, the probability of t_3 is $P(t_3|t_1, t_2)$.

We also wanted to explore the effect of the lexical emission order. For the default bigram emission order in HunPos, $k = 2$, the probability of w_2 is $P(w_2|t_1, t_2)$. Emission probability in TnT is fixed to $k = 1$.

In HunPos, there are also other possible parameters to tune, e.g. suffix length and rare word frequency for the handling of unknown words. For Swedish, however, changing these parameters have minor, if any, effect (Megyesi, 2008), so we used the default settings. And, unlike TnT, there are no smoothing parameters to tweak from the command line.

In the experiment, we therefore concentrated on the parameters for tag and emission order in the hidden Markov model. We used nodes (accessing maximally 4GB RAM during training[2]) in the UPPMAX computer grid[3], which could maximally

[2]Most nodes have a total of 8GB RAM, but not consecutive, so HunPos cannot use all of it.

[3]Uppsala Multidisciplinary Center for Advanced Computational Science. URL: `http://www.uppmax.uu.se/`.

train models for 5-grams for tag transition (tag order 4) and 4-grams for lexical emission (emission order 4). We varied both settings for both the source and the bootstrap corpus from 1 to 4, giving 256 combinations in all.

4 Results

Not surprisingly, as lexical models are larger than tag models and more so for large corpora, both memory usage and CPU time were mostly affected by emission order for the bootstrap corpus. The tagger with the widest view (4.4.4.4) maximally occupied 3.2GB, and took 1.5 hours to train and evaluate, while the tagger with the narrowest view (1.1.1.1) used a maximum of 0.5 GB and took 10 minutes. The 4.4.4.4 tagger also had the best overall accuracy of all taggers, 98.05%. The training phase requires more RAM than tagging. And although it takes a second or two to load the model before tagging starts, it is practically possible to use it in a computer with 2GB RAM. Furthermore, if the tagger is wrapped in a server, the model need only be loaded once.

Accuracy for the 2.2.2.2 (default) and 4.4.4.4 (best) tagger, respectively, is shown in Table 4, drilled-down by genre in SUC, and by known and unknown words. The 4.4.4.4 tagger overall improved .85 points over the 2.2.2.2 tagger. Most of the improvement lies in a better model for known words. For unknown words, on the other hand, the result is actually worse than for the 2.2.2.2 tagger. Forsbom (2008b) showed that genre composition of the bootstrap corpus had an effect on accuracy, both overall and drilled-down by genre. Here, we can see that the context size also matters. Fiction, for example, has above average overall accuracy with the 4.4.4.4 tagger, and below with the 2.2.2.2 one. Whether it has to do with a more formulaic language or not remains to be seen.

As the context size affects known and unknown words differently, we looked at the top 10 models for each of them. The ranking for the top 10 models for known words (see Table 6) follows the overall top 10 models (see Table 5) except for rank 9. The top 10 for unknown words (see Table 7) have only one model in common with the overall and known words top 10, namely rank 6, the 3.3.3.3 model. In a context where many unknown words are expected, the 3.3.3.3 model is a good compromise candidate.

To see the effect on accuracy of each setting, we used an Anova regression tree (Breiman et al., 1984; Therneau and Atkinson, 2004), where the accuracy for each combination is the response variable and each setting is a predictor variable. The regression tree was built using binary recursive partitioning of the data from the runs, where each split has a certain cost complexity. The cost complexity in combination with a cross-validation error, i.e. the "one standard-deviation rule" (Maindonald and Braun, 2003, p. 273f), was used to

	Settings				Accuracy		
Rank	SE	ST	BE	BT	Overall	Known	Unknown
1	4	4	4	4	98.05+0.10	98.50+0.08	85.28+1.03
2	3	4	3	4	97.97+0.10	98.41+0.07	85.51+1.03
3	4	4	3	4	97.96+0.09	98.39+0.07	85.38+0.99
4	3	4	4	4	97.93+0.11	98.37+0.08	85.15+1.07
5	4	3	4	3	97.89+0.12	98.32+0.08	85.54+1.02
6	4	4	4	3	97.88+0.11	98.32+0.10	85.26+1.01
7	4	3	4	4	97.86+0.13	98.30+0.10	84.95+0.98
8	3	3	3	3	97.83+0.11	98.24+0.09	85.78+0.98
9	3	3	4	3	97.81+0.11	98.23+0.09	85.57+0.96
10	4	3	3	3	97.80+0.11	98.23+0.10	85.62+0.99

Table 5: Top 10 models if sorted by overall accuracy. S=source model, B=bootstrapped model, E=emission order, T=tag order.

	Settings				Accuracy		
Rank	SE	ST	BE	BT	Overall	Known	Unknown
1	4	4	4	4	98.05+0.10	98.50+0.08	85.28+1.03
2	3	4	3	4	97.97+0.10	98.41+0.07	85.51+1.03
3	4	4	3	4	97.96+0.09	98.39+0.07	85.38+0.99
4	3	4	4	4	97.93+0.11	98.37+0.08	85.15+1.07
5	4	3	4	3	97.89+0.12	98.32+0.08	85.54+1.02
6	4	4	4	3	97.88+0.11	98.32+0.10	85.26+1.01
7	4	3	4	4	97.86+0.13	98.30+0.10	84.95+0.98
8	3	3	3	3	97.83+0.11	98.24+0.09	85.78+0.98
9	4	3	3	4	97.78+0.11	98.23+0.09	85.15+0.97
10	4	3	3	3	97.80+0.11	98.23+0.10	85.62+0.99

Table 6: Top 10 models if sorted by accuracy for known words. S=source model, B=bootstrapped model, E=emission order, T=tag order.

	Settings				Accuracy		
Rank	SE	ST	BE	BT	Overall	Known	Unknown
1	1	3	1	3	97.12+0.12	97.51+0.11	86.10+0.92
2	1	3	2	3	97.19+0.12	97.58+0.10	86.01+0.98
3	2	3	2	3	97.54+0.12	97.95+0.10	85.94+1.03
4	1	4	1	4	97.43+0.09	97.83+0.08	85.89+1.07
5	2	3	1	3	97.16+0.12	97.55+0.11	85.82+0.97
6	3	3	3	3	97.83+0.11	98.24+0.09	85.78+0.98
7	1	4	1	3	97.14+0.12	97.53+0.10	85.78+0.90
8	1	4	2	4	97.48+0.10	97.89+0.08	85.75+1.01
9	3	3	2	3	97.54+0.11	97.95+0.09	85.73+1.01
10	4	3	2	3	97.54+0.11	97.95+0.09	85.70+1.00

Table 7: Top 10 models if sorted by accuracy for unknown words. S=source model, B=bootstrapped model, E=emission order, T=tag order.

	2.2.2.2			4.4.4.4			
Genre	Overall	Known	Unknown	Overall	Known	Unknown	Prop. unknown
All	97.20±0.12	97.61±0.11	85.55±0.94	98.05±0.10	98.50±0.08	85.28±1.03	3.35±0.24
a	97.59±0.20	97.89±0.13	87.02±2.72	98.43±0.20	98.75±0.15	87.52±2.62	2.79±0.28
b	97.74±0.38	97.90±0.35	90.00±3.76	98.48±0.20	98.69±0.15	88.46±4.85	2.00±0.37
c	97.41±0.40	97.73±0.37	88.38±2.35	98.16±0.41	98.52±0.34	87.99±3.19	3.33±0.48
e	97.21±0.27	97.60±0.23	85.99±4.08	98.16±0.19	98.60±0.12	85.47±3.94	3.38±0.67
f	97.51±0.40	97.76±0.39	89.20±2.04	98.37±0.33	98.66±0.28	89.02±2.49	2.91±1.05
g	97.38±0.26	97.62±0.20	90.24±4.59	98.17±0.24	98.43±0.21	89.53±3.95	2.80±0.96
h	97.52±0.23	98.00±0.22	86.89±2.00	98.19±0.29	98.69±0.22	86.80±2.10	4.21±0.88
j	96.77±0.46	97.72±0.22	81.92±3.16	97.34±0.33	98.37±0.11	81.29±3.11	5.93±0.84
k	96.96±0.22	97.13±0.21	87.42±2.30	98.08±0.15	98.27±0.13	87.40±2.10	1.75±0.16

Table 4: Estimated accuracy and standard deviation for the 2.2.2.2 (default) and 4.4.4.4 (best) HunPos bootstrapped, drilled-down by SUC genre (10-fold cross-validation on SUC).

prune the resulting regression tree, to limit the risk of overfitting to the data. The rule says to prune a tree at the cost complexity of the first subtree with a cross-validation error larger than the minimal cross-validation error + 1 cross-validation standard deviation. For our overall tree, only one node was pruned.

The pruned regression tree, with a cross-validation error rate of 2%, is shown in Figure 1. As can be seen, the tag order plays the major role, both for the source and bootstrap corpora, and the main splits are between bigrams and trigrams. The setting used for the bootstrap corpus also seems more important than for the source corpus.

For unknown words, only the settings for tag order were used in building the regression tree. The tree, with one node pruned and a cross-validation error rate of as much as 8%, is shown in Figure 2. One thing that is more clear in the regression than when looking at the top 10 models is that for unknown words 4-grams seem optimal, while a wider context decreases the accuracy. In cases where many unknown words are expected, for example when moving to a new domain, it may therefore be wise to choose a lower tag order to get better results, whereas a good compromise could be the 3.3.3.3 model (cf. Tables 5–7).

As was the case for the top 10 models, the regression tree for known words (not included here) show a similar pattern to the overall tree. The only difference in tree structure is a missing subtree for known words, which corresponds to two nodes that were pruned from the known words tree.

5 Concluding remarks

In the paper, we looked at the effect of widening the context view, for tag transitions and lexical emissions, when bootstraping a raw corpus with a tagger trained on a validated source corpus.. Given current hardware limitations, we stopped at 5-grams (fourth order). A 5-gram hidden Markov model tagger, for example, gave better overall accuracy than a trigram tagger. Although memory requirements for training extend the average user's available RAM, tagging can be done in a reasonably equipped personal computer, even if loading the model takes time.

By means of a regression tree, we found in our experiment that a widened view was more important for tag transitions, and in particular for the bootstrap corpus. For lexical emission, it was also more important for the bootstrap corpus, although it was beneficial for both corpora. The main splits were between bigrams and trigrams. The best overall tagger was the one with the widest view for both tag transition and lexical emission, used for both corpora. It had an accuracy of 98.05%, compared to a bootstrapped tagger with only default settings, 97.20%. The improvement mainly occurred for known words, while the results for unknown words were actually worse. The optimal setting for unknown words was with 4-gram tag transition for both source and bootstrap corpora. The best compromise, if handling of unknown words is crucial, was the 3.3.3.3 model, 97.83%.

The widened view affected various genres in different degrees. Fiction, for example, benefited very much from it.

A selection of the models and accompanying information are available at `http://stp.lingfil.uu.se/~evafo/resources/taggermodels/`.

Acknowledgements

We would like to thank Anna Sågvall Hein and the anonymous reviewers for valuable comments,

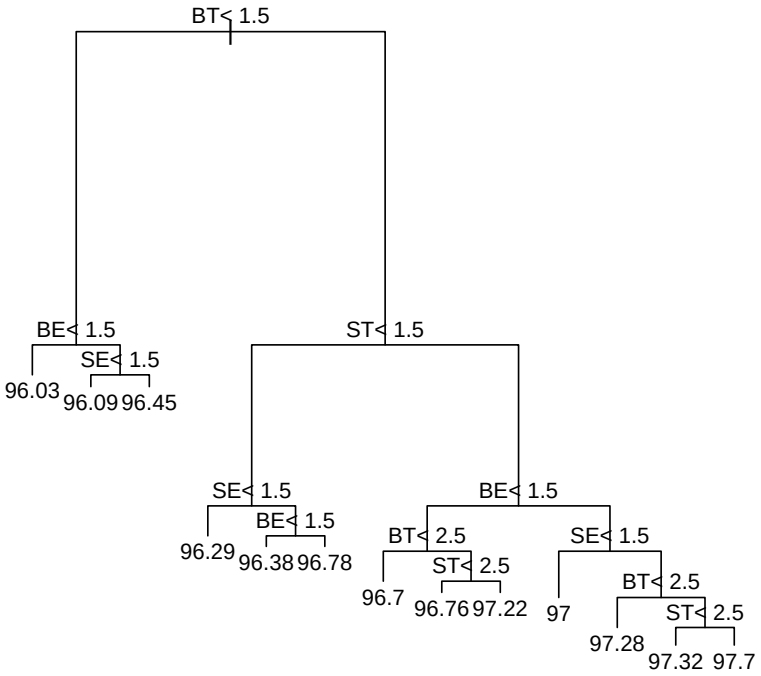

Figure 1: Regression tree for overall accuracy of bootstrapped models for various combinations of HunPos settings (10-fold cross-validation error rate=2%). S=source model, B=bootstrapped model, E=emission order, T=tag order.

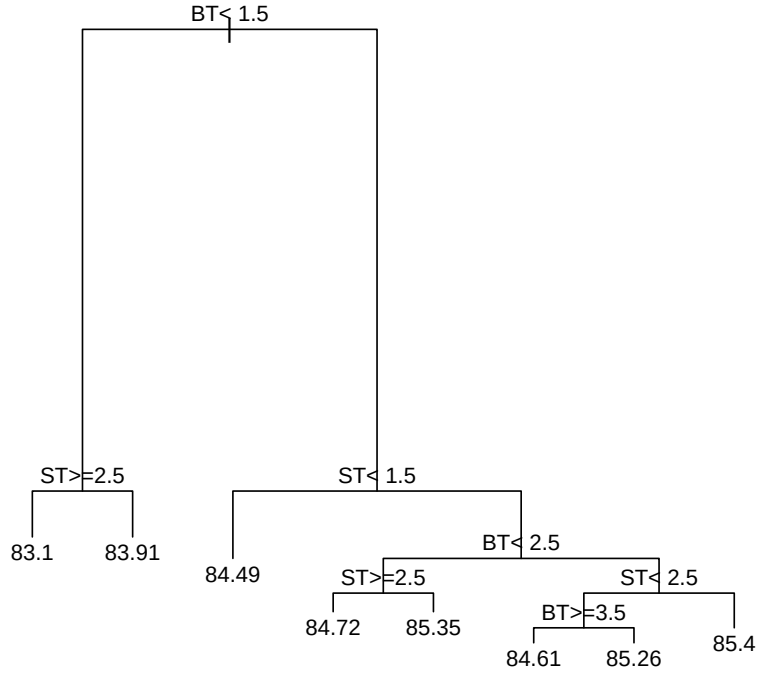

Figure 2: Regression tree for accuracy of unknown words in bootstrapped models for various combinations of HunPos settings (10-fold cross-validation error rate=8%). S=source model, B=bootstrapped model, E=emission order, T=tag order.

UPPMAX for letting us use their computer grid, Jonas Sjöbergh for a lexer which we based our PAROLE to Granska conversion script on, Johan Hall, Jens Nilsson and Joakim Nivre for a SUC version with Granska tags, which we used to tune the conversion script, and finally GSLT and NGSLT for funding.

References

Thorsten Brants. TnT - a statistical part-of-speech tagger. In *Proceedings of ANLP-2000*, Seattle, Washington, 2000.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, California, 1984.

Johan Carlberger and Viggo Kann. Implementing an efficient part-of-speech tagger. *Software Practice and Experience*, 29(9), 1999.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. The linguistic annotation system of the Stockholm–Umeå corpus project. Report DGL-UUM-R-33, Dep. of General Linguistics, University of Umeå, 1992.

Eva Ejerhed, Gunnel Källgren, and Benny Brodda. Stockholm-Umeå corpus version 2.0. Stockholm University, Dep. of Linguistics and Umeå University, Dep. of Linguistics, 2006.

Eva Forsbom. Big is beautiful: Bootstrapping a PoS tagger for Swedish, 2006. Poster presentation at GSLT retreat, Gullmarsstrand.

Eva Forsbom. Good tag hunting: Tagability of Granska tags. In *Resourceful Language Technology: Festschrift in Honor of Anna Sågvall Hein*. Uppsala University, 2008a.

Eva Forsbom. Size is not everything. genre balance in bootstrapping a Swedish PoS tagger. In *Proceedings of SLTC'08*, Stockholm, 2008b.

Péter Halácsy, András Kornai, and Csaba Oravecz. HunPos – an open source trigram tagger. In *Proceedings of ACL'07*, Prague, Czech Republic, 2007.

Johan Hall. A probabilistic part-of speech tagger with suffix probabilities. MSI report 03015, School of Mathematics and Systems Engineering, Växjö University, 2003.

John Maindonald and John Braun. *Data Analysis and Graphics Using R: An Example-based Approach*. Cambridge University Press, Cambridge, UK, 2003.

Beáta Megyesi. *Data-Driven Syntactic Analysis. Methods and Applications for Swedish*. Institution for Speech, Music and Hearing, Royal Institute of Technology, Stockholm, 2002.

Beáta Megyesi. The open source tagger HunPoS for Swedish. Report, Dep. of Linguistics and Philology, Uppsala University, 2008.

Bernard Merialdo. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2), 1994.

Joakim Nivre. Sparse data and smoothing in statistical part-of-speech tagging. *Journal of Quantitative Linguistics*, 7(1), 2000.

Joakim Nivre and Leif Grönqvist. Tagging a corpus of spoken Swedish. *International Journal of Corpus Linguistics*, 6(1), 2001.

Jonas Sjöbergh. Bootstrapping a free part-of-speech lexicon using a proprietary corpus. In *Proceedings of ICON-2003*, Mysore, India, 2003a.

Jonas Sjöbergh. Combining POS-taggers for improved accuracy on Swedish text. In *Proceedings of NoDaLiDa 2003*, Reykjavik, Iceland, 2003b.

Scott M. Thede and Mary P. Harper. A second-order hidden Markov model for part-of-speech tagging. In *Proceedings of ACL99*, College Park, Maryland, 1999.

Terry M. Therneau and Beth Atkinson. *rpart: Recursive Partitioning*, 2004. R package version 3.1-20. R port by Brian Ripley.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL'03*, Edmonton, Canada, 2003.

University of Gothenburg. The PAROLE corpus at The Swedish Language Bank. URL `http://spraakbanken.gu.se/parole/`.

ISSN 1736-6305 Vol. 4
http://hdl.handle.net/10062/9206