Prune Diseased Branches to Get Healthy Trees!
How to Find Erroneous Local Trees in a Treebank and Why It Matters

Markus Dickinson
Georgetown University

W. Detmar Meurers
The Ohio State University

1 Introduction and Motivation

Annotated corpora are essential for training and testing algorithms in natural language processing (NLP), but even so-called gold-standard corpora contain a significant number of annotation errors (cf. Dickinson 2005, and references therein). For part-of-speech annotation, these errors have been shown to be problematic for both training and evaluation of NLP technology (van Halteren 2000; Padro and Marquez 1998; van Halteren et al. 2001; Květǒn and Oliva 2002). But only little work has been done on detecting errors in syntactic annotation (Ule and Simov 2004; Dickinson and Meurers 2003b, 2005), and the effect of the errors detected on the uses of such corpora has not been systematically explored.

In this paper, we describe a new method for finding errors in treebanks and demonstrate the effect of such errors on NLP technology. Similar to the work in Dickinson and Meurers (2003b, 2005)—where multiple occurrences of the same string in identical contexts are found with varying labels—the approach presented here is based on the detection of inconsistencies; but instead of focusing on the consistent assignment of a label to a string, we here investigate the consistency of labeling within local trees. In section 2, we describe the new method, and we show the results of applying it to the Wall Street Journal corpus in section 3. After discussing ways to automatically identify individual erroneous rules in section 4, we show in section 5 that eliminating the detected errors from the training data of a probabilistic context-free grammar (PCFG) parser improves its performance.

2 A new error detection method

Most natural language expressions are analyzed as endocentric, i.e., a category projects to a phrase of the same general category. For example, an adjective will project to an adjectival phrase. This assumption is directly encoded in the widely adopted X-bar schema (Jackendoff 1977), a generalization over phrase structure rules, and similar generalizations are encoded in virtually all constituency-based syntactic frameworks. An interesting effect of this organization of constituent structure is that one can generally determine the syntactic category of the mother based on the categories of the daughters. We thus propose to systematically search for variation in the mother categories dominating the same daughters in order to find erroneous annotation in local trees.

More concretely, to identify potentially erroneous rules, we extract all local trees from a treebank and index them by the daughters list. For each list of daughters (consisting of part-of-speech labels for lexical daughters and syntactic category labels for phrasal daughters) the set of immediately dominating mothers is determined. If this *immediate dominance (ID) set* has more than one element, we will say that a daughters list shows *ID variation* and interpret it as an indication of a potential error. Note that this approach turns the usual conceptualization of local trees as rules on its head: instead of looking for which daughters can expand a fixed mother category, we are fixing the daughters and asking which mother categories can dominate these daughters.

Let us illustrate the idea with an example from the Wall Street Journal (WSJ) as annotated in the Penn Treebank 3 (Marcus et al. 1993). The daughters list ADVP VBN NP (adverbial phrase, past participle, noun phrase) occurs 167 times in the WSJ, with two distinct mother categories: VP (verb phrase, 165 times) and PP (prepositional phrase, 2 times). Based on the endocentricity considerations mentioned above, a past participle verb (VBN) is expected to project to a verbal category, such as VP, but not to a PP—and indeed the two trees dominated by PP turn out to be incorrect.

The same underlying idea is reflected in projects such as the CCGbank (Hockenmaier and Steedman 2005), a categorial grammar style annotation derived from the Penn Treebank, which fixed some errors present in the original treebank. For this purpose, specific rules such as the following are written to identify and fix particular patterns (Hockenmaier and Steedman 2005, p. 94): "Under ADVP, if the adverb has only one child, and it is tagged as NNP, change it to RB. This is a tagging error, and we do not want unknown NNPS to have adverbial categories." Likewise, Blaheta (2002) deals with hard cases, such as the distinction between preposition (IN), adverb (RB), and particle (RP), using rules of the form "If an IN is occurring somewhere other than under a PP, it is likely to be a mistag." While the rules used in these approaches reflect the assumption of endocentricity that our approach is based on, we propose to search for all such endocentricity violations, and our method detects such violations automatically, instead of being based on human inspiration encoded in hand-written rules.

3 ID variation in the WSJ and the errors it points to

Running the algorithm sketched above on the entire WSJ corpus as annotated in the Penn Treebank 3, we obtain 844 distinct lists of daughters which are assigned more than one mother category in the corpus.

To investigate how many of these variations in mother category point to errors, we randomly sampled 100 of these daughters lists and manually evaluated for each whether it points to an error, a genuine ambiguity, or whether it was unclear according to the annotation guidelines. We only count a daughters list as pointing to an error if for (at least) one of the mothers in the ID set, every occurrence of the daughters with that mother is an incorrect annotation. Of the 100 daughters lists, 74 pointed to an error, 24 were genuine ambiguities, and 2 were unclear. We therefore estimate that 74% (95% CI[1]: 65.4% to 82.6%) of the 844 daughters lists, i.e., 625 (552 to 697) daughters lists, point to errors.

Since each pairing of a daughters list and a mother category, i.e., each distinct local tree, constitutes an instance of a phrase structure rule, it is interesting to know how many rules the 100 daughters lists with their possible mother categories correspond to. For our sample of 100 daughters lists, there are 291 such rules, i.e., an average of 2.91 mother categories per ID set. For the full set of 844 daughters lists, there are 2201 rules.

Zooming in on the 74 cases pointing to at least one erroneous rule, we want to know how many of the elements of the ID set are errors (it could be anywhere between one and the entire set, in which case none of the annotations are correct). The 74 daughters lists correspond to 223 different rules; examining all instances shows that 127 of the rules are errors. For example, the daughters list IN NP (preposition/subordinating conjunction, noun phrase) has a nine element ID set (i.e., nine different mothers in the corpus). Three of the elements (PP, FRAG, X) can indeed occur as mothers for the daughters list IN NP; the six others (ADJP, ADVP, NP, SBAR, VP, WHPP) are never correct. The daughters list IN NP thus points to six erroneous rules. Based on 127 of 291 rules in our random sample being errors, we estimate 43.6% (95% CI: 37.9% to 49.4%) of all rules in our variations to be errors, i.e., 961 (834 to 1087) of the 2201 rules.[2]

Analyzing the nature of the 74 daughter lists pointing to errors, the errors fall into three groups: daughter label errors (38), mother label errors (41), and bracketing errors (13), with some errors having multiple causes. For example, take the daughters list IN NP when it is dominated by VP. In (1a), we see a bracketing error,

[1] Confidence Interval (Fleiss 1981).

[2] Because the data for rules is correlated within daughters (intracluster correlation = 0.01), the standard error used in this confidence interval is adjusted by a variance inflation factor (design effect) (cf. Fleiss 1981; Donner and Klar 2000).

where all of *runs*, *up*, and *high commission costs* should have been bracketed as daughters of a mother VP. (The part-of-speech for *up* is also wrong; it should be particle (RP).) In (1b), we see a mother category error: *past it* is not a verb phrase, but a prepositional phrase (or perhaps adverbial). And in (1c), we find a daughter error in the tagging of *like* as a preposition (IN) when it is a verb (VB).

(1) a. Frequent trading runs [$_{VP}$ up/IN [$_{NP}$ high commission costs]] .
b. Turkey in any event is long [$_{VP}$ past/IN [$_{NP}$ it]] .
c. Mr. Friend's client [. . .] didn't [$_{VP}$ like/IN [$_{NP}$ the way 0 defense attorney Tom Alexander acted during the legal proceedings *T*]] .

In addition to examining the number of types of local trees affected, we also counted the number of local tree tokens which are error instances of the 100 ID set sample. Of the 127 tree types (rules) which are erroneous, we find 847 occurrences in the treebank.

3.1 Treebank issues

Given that our method is based on a linguistic consideration, it is useful to discuss how the nature of the particular annotation scheme underlying the Penn Treebank affects the precision of our error detection method.

Null elements In the Penn Treebank, null elements have been inserted into the text so that certain phrasal categories retain their daughters even if they are not overtly realized. At first glance, corpus annotation preserving daughter information in this way seems to be useful for an error detection method like ours, which is based on pairs of daughters lists and mothers. However, since our method starts out from the daughters lists and compares which mothers can dominate a given daughters list, the advantage of an annotation scheme including empty daughters is relatively small and solely derives from the fact that including null daughters in the annotation makes it possible to compare the mother categories assigned to more instances of a given daughters list. Our approach thus is equally applicable to treebanks which maintain a closer connection between the syntactic annotation and the actual, overt surface realization.

Annotation violating endocentricity Some properties of the distinctions made in the WSJ treebank cause problems for a method searching for endocentricity violations, namely aspects of the annotation scheme which directly violate endocentricity. The guidelines for proper nouns (words annotated with the part-of-speech tags NNP or NNPS) are a good example of this. The rule for part-of-speech annotation given in Santorini (1990, p. 32) is that capitalized words which appear in a title should be tagged NNP, as in example (2).

(2) A/NNP Tale/NNP of/IN Two/NNP Cities/NNP

In the syntactic annotation guidelines (Bies et al. 1995, p. 207), however, titles are specified to be annotated like running text (and receive a TTL (title) function tag), as illustrated in (3).[3] As a result, the title *Saved By The Bell* is annotated in the Penn Treebank as shown as in (4).

(3) [$_{S-TTL}$ [$_{NP-SBJ}$ *] [$_{VP}$ Driving [$_{NP}$ Miss Daisy]]]

(4) [$_{NP-TTL-PRD}$ [$_{S}$ [$_{NP-SBJ}$ *] [$_{VP}$ Saved/NNP [$_{PP}$ By/NNP [The/NNP Bell/NNP]]]]

For our discussion, the key part of (4) is the annotation of *Saved* and the VP dominating it. Following the part-of-speech guidelines, *Saved* is annotated as nominal (NNP) since it is part of a capitalized title. But the syntactic annotation implicitly views *Saved* as a verb, so that it is annotated as part of a VP.

In sum, the mismatch between part-of-speech and structural syntactic annotation results in a violation of endocentricity. Our method will thus flag such examples as potentially erroneous, even though they are in accordance with the guidelines—but one can argue that such a mismatch between the part-of-speech and the syntactic guidelines is indeed something that would deserve to be revisited.

4 From ID variation to automatic detection of erroneous rules

In the previous section, we established that ID variation is useful for finding incorrectly annotated local trees and, by extension, the rules licensing these trees. To make this observation practically useful for a large corpus, where hand validation of all ID variation may not be feasible, we want to define a heuristic for automatically detecting which of the elements in the ID set of a given daughters list are errors and which are a part of a legitimate ambiguity. In other words, we have to prune the set of acceptable treebank rules, akin to the task of post-pruning in machine learning (e.g., Fürnkranz 1997).

[3] The title is analyzed as including an empty subject NP, so that the TTL function tag marking the expression as a title is not attached to the VP, but the overall S.

4.1 Frequency-based heuristics

Absolute number of rule occurrences For PCFG parsing, it is often assumed that one can prune low-frequency rules without a degradation in parsing performance (Gaizauskas 1995; Charniak 1996; Cardie and Pierce 1998) (although, for certain kinds of trees, keeping low-frequency rules has been shown to improve performance (Bod 2003)). The underlying idea is that rules which occur rarely are more likely to have been a mistake, or if not a mistake, the probability of their occurrence is so small as to be negligible. Based on this idea, one can create a heuristic classifying the low-frequency ID variation categories as errors and the frequent ones as genuine ambiguities.

We first isolated all rules in the ID sets which occurred only once, in order to gauge how well this simple and commonly-used method of elimination works. The results are given in Figure 1, for both type and token counts. We report how many of the erroneous trees the heuristic identifies out of all the trees it identifies (precision) and out of all the trees it should have identified (recall). Since each rule we detect here occurs only one time, the type and token precision figures are exactly the same.

	Precision	Recall
Types	74.75% (74/99)	58.27% (74/127)
Tokens	74.75% (74/99)	8.74% (74/847)

Figure 1: Evaluating the absolute frequency detection heuristic

Figure 1 shows high precision in that most single-occurrence rules are indeed erroneous; but, because these rules occur once, the token recall is very low. There are two main reasons that this predictor is insufficient. On the one hand, we find examples of frequently-occurring rules which are incorrect, such as the rule NP → VBG which appears 177 times, despite being wrong. On the other hand, there are examples of infrequently-occurring rules which are correct; e.g., S → NP S is correct even though it occurs only once, in comparison to the same daughters occurring 393 times with NP as the mother. Of the 99 rules in our set which occur once, a full 25 of them are correct.

Relative token frequency within ID sets The first heuristic considers only the token counts and does not use information found within the variations, i.e., the ID sets. To identify more rule occurrences which are wrong, we can use the properties found in a variation to define a second frequency-based heuristic.

For each variation we took the total number of token occurrences of the daughters list and extracted all rules whose token occurrences were less than 10% of the total number of instances with this daughters list. For example, there are 86 total times in the corpus where the daughters list NNP CC NNP NNP appears bracketed as a constituent. NP (noun phrase) is the mother 83 times and thus is likely correct, but UCP (unlike coordinated phrase) occurs only twice as mother of this daughters list, or in 2.33% of the cases. Thus, UCP is flagged as a likely error, and indeed it is erroneous.

As we can see in Figure 2, detecting erroneous rules with this 10% metric obtains approximately 60% precision and 60% recall on types. But the token precision results are significantly worse.

	Precision	Recall
Types	60.47% (78/129)	61.42% (78/127)
Tokens	9.20% (499/5424)	58.91% (499/847)

Figure 2: Evaluating the relative frequency heuristic

Continuing with the NNP CC NNP NNP example, for instance, we find one occurrence of NX as a mother, but this is correct. Having more of an impact, we find rules under the 10% threshold despite being quite frequent; for example, the label NX (certain complex noun phrases) appears 102 times as the mother of JJ NN, but NP appears 5972 times as the mother of the same daughters, so the correct label NX is ruled out by being under the 10% mark. Even if we remove the single greatest occurrence below the 10% threshold, the 4297 occurrences of NP → NP VP, the token precision is still only 44.28%. In sum, the second frequency-based heuristic dramatically improves the recall, but for the reasons just mentioned, the token precision suffers equally dramatically.

There are other possible frequency-based methods one can consider, such as using a 10% threshold combined with an additional absolute threshold, but preliminary results show this to be similarly unsuccessful. Frequency-based heuristics can get high precision or high recall for the task of automatically detecting erroneous rules, but not both. Infrequent correct rules and frequent incorrect rules both cause problems for frequency-based methods.

4.2 Adding an ambiguity measure

Instead of relying solely on low token frequency, we need another property which makes detecting erroneous rules more accurate. But what kind of properties can we use? To answer this question, consider the following example: regardless of the number of token occurrences within a given variation, NP and NX can generally vary between one another, since their distribution is dependent on the linguistic material outside the constituent. The label NX is used for noun phrases which share a modifier with another noun phrase; if there is no shared modifier, then the identical-looking constituent is labeled NP. So both categories label nominals, and both are likely to occur with the same list of daughters.

To address such cases, we combine a token frequency measure with a measure of how likely it is for two categories to be involved in a genuine ambiguity. We start out with the token frequency measure evaluated in Figure 2, where for each variation we take the total number of occurrences of a daughters list and mark as errors all ID set elements whose token occurrences make up less than 10% of the occurrences in their ID set. Then we restrict the set of potential errors identified this way by eliminating all ID set categories which, when paired with the most frequent category in the ID set, are among the top five variations in the corpus. For example, for the daughters JJ NN, because the mother label NX varies with NP throughout the corpus and the two are among the five most frequent mother categories occurring together in ID variation sets, it is not flagged as an error.

For our sample of 100 ID variations, the resulting precision and recall figures for automatically detecting which variations are errors are shown in Figure 3.

	Precision	Recall
Types	73.03% (65/89)	51.18% (65/127)
Tokens	65.59% (364/555)	42.98% (364/847)

Figure 3: Combining the relative frequency heuristic with an ambiguity measure

The combined heuristic results in an error detection precision approaching that of the single occurrence heuristic we saw in Figure 1, while recovering nearly half of the errors. We see an increase in precision from Figure 2 without a severe drop in recall because the ambiguity measure lets us sort out highly frequent rules based on something other than frequency. For instance, in the example of the daughters list JJ NN, ADJP occurs 25 times as a mother but less than 10% of the time within the variation; it would thus be incorrectly flagged as an error by the 10% heuristic. However, because the pairing ADJP–NP is the most frequent overall ambiguity, the rule is correctly not flagged as an error by the combined frequency/ambiguity heuristic. On the other hand, with the daughters list IN NP, the mother ADVP occurs 170 times, but the pairing ADVP-PP is not one of the five most frequent, so this is still flagged as an error, and correctly so.

While this heuristic for spotting erroneous rules in ID variation sets can clearly be improved, the results are encouraging enough to try to measure the impact of removing all rules detected by this method. At the same time, the above experiments also showcase the difficulty of deciding automatically whether a given rule used in a treebank is correct or not.

5 Impact of erroneous rule elimination on PCFG parsing

To test the impact of erroneous rules on PCFG parsing, we used the left-corner parser LoPar (Schmid 2000),[4] which has the nice property that, after training, one can remove rules from the set of rules with their frequency counts. We split the WSJ corpus into training (sections 2-21) and test (section 23) data. We compared using all grammar rules from sections 2-21 of the treebank (*All*, 15,246 rules) with using the set of rules from which we removed all rules flagged by the combined frequency/ambiguity heuristic discussed in the previous section (*Reduced*, 14,798, i.e., 448 rules were removed).[5]

The parsing results using these two PCFG models on the test data are given in Figure 4, using the standard PARSEVAL measures (Black et al. 1991), i.e., bracketing precision, recall, and F-measure, for both labeled and unlabeled evaluation.[6]

	Precision		Recall		$F_{\beta=1}$	
	Lab.	Unl.	Lab.	Unl.	Lab.	Unl.
All	70.39%	74.73%	67.31%	71.46%	68.82%	73.06%
Reduced	71.48%	75.68%	68.40%	72.42%	69.91%	74.01%

Figure 4: LoPar results using full and reduced rule set

The figure shows an improvement in precision and recall for the reduced rule set. To see whether the improvement was due to chance or not, we computed the significance of the precision and recall changes using stratified shuffling and found that the changes are significant at the $\alpha = 0.001$ level.[7] In comparison, removing low-frequency rules (Gaizauskas 1995; Charniak 1996; Cardie and Pierce 1998) by eliminating all rules which occur only once resulted in some improvement over the *All* baseline, but not significantly in the way our method does.

We conclude that the presence of erroneous rules in a grammar induced from a treebank is harmful for parsing precision and recall and that targeting and eliminating erroneous rules can improve parser performance.

[4] We used the unlexicalized, non-headed version. LoPar is available from `http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/LoPar-en.html`

[5] Note that all rules flagged by the heuristic were removed, which includes some correct rules.

[6] Scores were computed using "evalb" (`http://nlp.cs.nyu.edu/evalb/`) by Satoshi Sekine and Michael John Collins.

[7] This was carried out using Dan Bikel's Randomized Parsing Evaluation Comparator (`http://www.cis.upenn.edu/~dbikel/software.html`).

6 Summary and Outlook

We presented a new method for detecting bracketing and labeling errors in syntactic annotation and demonstrated its effectiveness for the WSJ treebank. The method is inspired by the linguistic concept of endocentricity and its consequence that the list of daughters in a local tree constrains the possible categories of the mother in that local tree. To determine which mother node variations are likely to be errors, we explored several heuristics and demonstrated that frequency by itself is an insufficient predictor of errors. We instead proposed a heuristic combining frequency with an expected-ambiguity measure and show that removing the rules thus flagged as errors from the set of rules used by a PCFG parser leads to an improved performance of the parser.

In the future, we intend to explore more complex heuristics to improve precision/recall of errors and to determine the exact effect of the treebank errors on the PCFG parser. Given the well-known problems with the standard PARSEVAL measures (e.g. Carroll et al. 2002), we would also like to explore an evaluation with other methods for comparing parser output. Along those lines, we would like to perform a more robust error analysis, to determine the kinds of differences between the parser output and the benchmark corpus, and to determine whether any parsing errors are actually treebank errors. Finally, to explore the applicability of the method, we intend to test it on other treebanks with different annotation schemes.

Acknowledgements We would like to thank the participants of the OSU discussion group CLippers for encouraging feedback on this topic, and we are extremely grateful to Stephanie Dickinson for lending her statistical expertise.

References

Bies, A., M. Ferguson, K. Katz and R. MacIntyre (1995). *Bracketing Guidelines for Treebank II Style Penn Treebank Project*. University of Pennsylvania.

Black, E., S. Abney et al. (1991). A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*. Pacific Grove, CA: Morgan Kaufmann.

Blaheta, D. (2002). Handling noisy training and testing data. In *Proceedings of the 7th conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 111–116.

Bod, R. (2003). Do All Fragments Count? *Natural Language Engineering* 9(4), 307–323.

Cardie, C. and D. Pierce (1998). Error-driven pruning of Treebank grammars for base noun phrase identification. In COLING/ACL (1998), pp. 218–224.

Carroll, J., A. Frank, D. Lin, D. Prescher and H. Uszkoreit (eds.) (2002). *Proceedings of the Workshop "Beyond PARSEVAL. Towards Improved Evaluation Measures for Parsing Systems" at the 3rd International Conference on Language Resources and Evaluation (LREC-02)*. Las Palmas, Gran Canaria.

Charniak, E. (1996). Tree-Bank Grammars. In *AAAI/IAAI, Vol. 2*. pp. 1031–1036.

COLING/ACL (1998). *Proceedings of the 17th International Conference on Computational Linguistics (COLING) and the 36th Annual meeting of the Association for Computational Linguistics (ACL)*. Montreal.

Dickinson, M. (2005). Error detection and correction in annotated corpora. Ph.D. thesis, The Ohio State University.

Dickinson, M. and W. D. Meurers (2003a). Detecting Errors in Part-of-Speech Annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*. Budapest, pp. 107–114.

Dickinson, M. and W. D. Meurers (2003b). Detecting Inconsistencies in Treebanks. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories (TLT-03)*. Växjö, Sweden, pp. 45–56.

Dickinson, M. and W. D. Meurers (2005). Detecting Errors in Discontinuous Structural Annotation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*. Ann Arbor, pp. 322–329.

Donner, A. and N. Klar (2000). *Design and Analysis of Cluster Randomization Trials in Health Research*. London: Arnold.

Fleiss, J. L. (1981). *Statistical Methods for Rates and Proportions*. New York: John Wiley and Sons, Inc.

Fürnkranz, J. (1997). Pruning Algorithms for Rule Learning. *Machine Learning* 27(2), 139–171.

Gaizauskas, R. (1995). *Investigations into the grammar underlying the Penn Treebank II*. Tech. Rep. Research Memorandum CS-95-25, University of Sheffield.

Hockenmaier, J. and M. Steedman (2005). *CCGbank: User's Manual*. Tech. Rep. MS-CIS-05-09, Department of Computer Science and Information Science, University of Pennsylvania, Philadelphia.

Jackendoff, R. (1977). *X-bar Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT Press.

Klein, D. and C. D. Manning (2001). Parsing with Treebank Grammars: Empirical Bounds, Theoretical Models, and the Structure of the Penn Treebank. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-01)*. Toulouse, pp. 330–337.

Krotov, A., M. Hepple, R. J. Gaizauskas and Y. Wilks (1998). Compacting the Penn Treebank Grammar. In COLING/ACL (1998), pp. 699–703.

Květŏn, P. and K. Oliva (2002). Achieving an Almost Correct PoS-Tagged Corpus. In P. Sojka, I. Kopeček and K. Pala (eds.), *Text, Speech and Dialogue 5th International Conference (TSD)*. Heidelberg: Springer, pp. 19–26.

Marcus, M., B. Santorini and M. A. Marcinkiewicz (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2), 313–330.

Padro, L. and L. Marquez (1998). On the Evaluation and Comparison of Taggers: the Effect of Noise in Testing Corpora. In COLING/ACL (1998), pp. 997–1002.

Santorini, B. (1990). Part-Of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision, 2nd printing). Ms., University of Pennsylvania.

Schmid, H. (2000). *LoPar: Design and Implementation*. No. 149 in Arbeitspapiere des Sonderforschungsbereiches 340 (SFB 340). Stuttgart: IMS, Universität Stuttgart.

Ule, T. and K. Simov (2004). Unexpected Productions May Well be Errors. In *Proceedings of 4th Int. Conference on Language Resources and Evaluation (LREC-04)*. Lisbon, Portugal.

van Halteren, H. (2000). The Detection of Inconsistency in Manually Tagged Text. In A. Abeillé, T. Brants and H. Uszkoreit (eds.), *Proceedings of the Second Workshop on Linguistically Interpreted Corpora (LINC-00)*. Luxembourg.

van Halteren, H., W. Daelemans and J. Zavrel (2001). Improving Accuracy in Word Class Tagging through the Combination of Machine Learning Systems. *Computational Linguistics* 27(2), 199–229.