In: Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003). Växjö, Sweden.

Detecting Inconsistencies in Treebanks

Markus Dickinson and W. Detmar Meurers

The Ohio State University
Department of Linguistics
E-mail: {dickinso,dm}@ling.osu.edu

1 Introduction

Treebanks have at least two uses: Firstly, as data for theoretical linguists to search through for theoretically relevant patterns, and secondly as training and "gold standard" testing material for computational linguists developing parsers and other language technologies. Treebanks are generally the result of a manual or semi-manual mark-up process. They thus can contain annotation errors from automatic preprocesses, human post-editing, or human annotation. The presence of errors can cause problems for both theoretical and computational linguists, making for low precision and recall of queries for already rare linguistic phenomena, and for less reliable training and evaluation of natural language processing technology. Investigating the quality of treebank annotation and improving it where possible is therefore desirable both from a theoretical and a computational linguistic perspective.

Given that the creation of treebanks typically involves a manual or semi-manual process, the lack of consistency of humans is an important potential source of annotation errors. The idea that variation in annotation can indicate annotation errors has recently been explored to detect errors in part-of-speech (pos) annotation (van Halteren, 2000; Eskin, 2000; Dickinson and Meurers, 2003). The method for detecting variation in pos-annotation we proposed in Dickinson and Meurers (2003) is related to the common interannotator agreement evaluation (cf., e.g., Brants and Skut, 1998) which compares multiple annotations of the same sentence at the same corpus position. Our inconsistency detection approach differs, however, from the interannotator agreement approach in that it compares the occurrence of identical words with similar contexts throughout a single annotated version of the corpus. In this paper, we discuss how one can extend our approach to the detection of errors in syntactic annotation.

2 Starting point: Variation detection for pos-annotation

In part-of-speech tagging, there is a lexically determined set of tags that can in principle be assigned to each word occurring in a corpus. The tagging process reduces this set of lexically possible tags to the correct tag for a specific corpus occurrence. A particular word occurring more than once in a corpus can thus be assigned different tags in a corpus. This is what in Dickinson and Meurers (2003) we refer to as *variation*. Such variation in corpus annotation is caused by one of two reasons: i) *ambiguity*: there is a word ("type") with multiple lexically possible tags and different corpus occurrences of that word ("tokens") happen to realize the different options,[1] or ii) *error*: the tagging of a word is inconsistent across comparable occurrences.

The main idea behind the proposal in Dickinson and Meurers (2003) is to to detect so-called *variation n-grams*. These variation n-grams are recurring stretches of text with variation in the annotation, the source of the variation being the so-called *variation nucleus*. The variation nucleus is a word which has different taggings despite occurring in the same context, in this case surrounded by identical words. For example, in the Wall Street Journal (WSJ) corpus (Taylor et al., 2003), the string in (1) is a variation 12-gram since *off* is a variation nucleus that in one corpus occurrence of this string is tagged as preposition (IN), while in another it is tagged as a particle (RP).

(1) to ward off a hostile takeover attempt by two European shipping concerns

To use this idea in practice for the detection of pos-annotation errors, Dickinson and Meurers (2003) propose an efficient way of computing all variation n-grams for a corpus and define heuristics for deciding whether a particular variation is an ambiguity or an error. The variation n-grams are calculated using an algorithm which essentially is an instance of the a priori algorithm used in information extraction (Agrawal and Srikant, 1994). It obtains all variation n-grams from $n = 1$ to the longest n for which there still is variation in the corpus. The algorithm finds the variation nuclei and works outward to the right and left in order to find longer and longer stretches of text which still contain a nucleus with variation in its tagging. By filtering these variation n-grams with two heuristics—trust variation nuclei with long contexts, distrust variation on the fringe of a variation n-gram—the method is shown to detect a large number of tagging errors in the Wall Street Journal corpus with high precision.

[1] For example, the word *can* is ambiguous between being an auxiliary, a main verb, or a noun and thus there is variation in the way *can* would be tagged in *I can play the piano*, *I can tuna for a living*, and *Pass me a can of beer, please*.

3 Detecting variation in syntactic annotation

3.1 Defining variation nuclei for syntactic annotation

To adapt the variation n-gram method for the detection of errors in syntactic annotation, we must define what constitutes a nucleus as the unit of data for which we compare annotations. For Dickinson and Meurers (2003), single words (tokens) were the unit of data and each word was paired with a part-of-speech tag as the annotation we were interested in comparing. For syntactic annotation it is not as straightforward to determine a unit of data with a one-to-one relation to the syntactic category annotation we want to compare since the syntactic category labels annotate constituents, which are strings (token lists) of different length. In other words, the length of the string making up a constituent is determined by the annotation, but we are looking for a theory-independent, data-driven definition of the nuclei for which the annotation is compared.

As a solution to this problem, we decompose the variation n-gram detection for syntactic annotation into a series of runs with different nucleus sizes. Each run detects the variation in the annotation of strings of a specific length. By performing such runs for strings from length 1 to the length of the longest constituent in the corpus we ensure that all strings which are analyzed as a constituent somewhere in the corpus are compared to the annotation of other occurrences of that string.

There are two things worth pointing out about this method for comparing syntactic annotation: Firstly, when comparing the annotation of a string of a specific length, only the category assigned to that entire string is compared. The internal structure of a constituent, i.e., the syntactic annotation of its substrings, is inspected when the nuclei of the length of the respective substring are compared, i.e., during a different run. Secondly, since we organize the variation detection as a data-driven search from strings of a particular length to the syntactic categories assigned to those complete strings, we need to decide how to handle the case in which a string has an occurrence in which it is not analyzed as a constituent and therefore not assigned a syntactic category. To handle those cases, we assign all non-constituent occurrences of a string the special label NIL. Note that this has the effect that a string occurring multiple times in the corpus without ever being annotated as a constituent will not show up as variation.[2]

[2] Our method therefore does not detect differences in constituents which have one but not both constituency borders within the nucleus, as e.g., the variation between $A[B\underline{\text{X Y]}}C$ and $D[E\underline{\text{X] Y}}F$ or $[DE\underline{\text{X Y]}}F$ (with the nucleus shown in the underlined gray box). One could explore using a range of special labels to encode more information about those differences, but we do not pursue this here since it violates the general idea of our approach to compare the annotation of identical recurring nuclei given that A, B, and C differ from D, E, and F (if they are identical, the variation *is* caught by our approach when the nuclei $\underline{\text{B X Y}}$, $\underline{\text{E X}}$, and $\underline{\text{D E X Y}}$ are investigated).

Let us take a look at an example from the WSJ treebank, the variation 12-gram in (2), which includes a nucleus of size two.

(2) market received its biggest jolt last month from Campeau Corp. , which

The string *last month* is a variation nucleus in this 12-gram because in one instance in the corpus it is analyzed as a noun phrase (NP), as in Figure 1 while in another it does not form a complete constituent on its own, as shown in Figure 2.

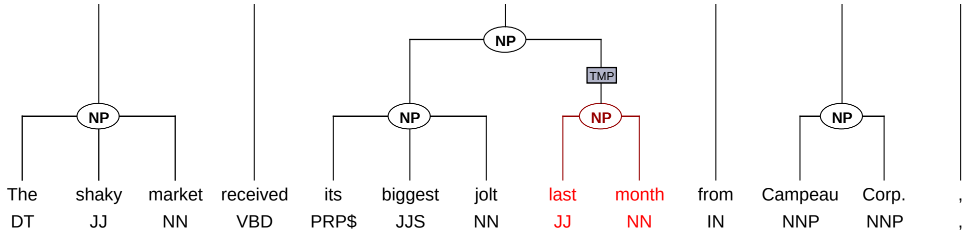

Figure 1: An occurrence of “last month” as a constituent

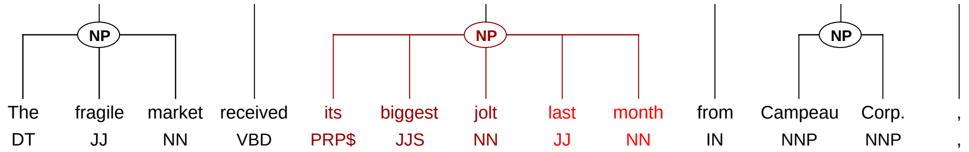

Figure 2: An occurrence of “last month” as a non-constituent

As another example, take the variation 4-gram *from a year earlier*, which appears 76 times in the WSJ. Out of those, the nucleus *a year* is labeled noun phrase (NP) 68 times, and 8 times it is not annotated as a constituent. Finally, an example with two syntactic categories involves the nucleus *next Tuesday* as part of the variation 3-gram *maturity next Tuesday*, which appears three times in the WSJ. Twice it its labeled as a noun phrase (NP) and once as a prepositional phrase (PP).

Having clarified how we can (re)define the notion of a variation nucleus to detect variation in syntactic annotation, we now take a closer look at how the variation nuclei and the variation n-grams are computed for a given treebank.

3.2 Computing the variation nuclei of a treebank

A simple way of calculating all variation nuclei would be to perform the following steps for all i between 1 and the length of the longest constituent in the corpus: First, step through the corpus and store all stretches of length i with their category label, or with the special label NIL if that list of corpus positions is not annotated to be a constituent. Second, eliminate the non-varying ones.

However, such a generate-and-test methodology which for every corpus position stores the list of words of length i for all i up to the length of the longest constituent in the corpus is clearly not feasible for dealing with larger corpora. Instead, we can make use of the observation from the last section that a variation necessarily involves at least one constituent occurrence of a nucleus. We thus arrive at the following algorithm to calculate the set of nuclei for a window of length $i(1 \leq i \leq \textit{length-of-longest-constituent-in-corpus})$:

1. Compute the set of nuclei:
   a) Find all constituents of length i, store them with their category label.
   b) For each distinct type of string stored as a constituent of length i, add the label NIL for each non-constituent occurrence of that string.
2. Compute the set of variation nuclei by determining which of the nuclei were stored in step 1 with more than one label.

In addition to calculating the variation nuclei in this way, we generate the variation n-grams for these variation nuclei as defined in Dickinson and Meurers (2003), i.e., we search for instances of the variation nuclei which occur within a similar context. The motivation for the step from variation nuclei to variation n-grams is that a variation nucleus occurring within a similar context is more likely to be an error. Regarding what constitutes a similar context, for the purpose of this paper we simply define it as a context of identical words surrounding the variation nucleus, just as in Dickinson and Meurers (2003). To increase the recall of the error detection method one could experiment with using a less strict notion of similarity, such as requiring the context surrounding the nucleus to consist of identical or related pos-tags instead of identical words.

4 A case study: Applying the method to the WSJ

To test the variation n-gram method as applied to syntactic annotation, we did a case study with the already mentioned WSJ corpus as part of the Penn Treebank 3 (Marcus et al., 1999). Before presenting the results of the case study, there are a couple of points to note about the nature of the corpus and the format we used it in.

Syntactic categories and syntactic functions The syntactic annotation in the WSJ includes syntactic category and syntactic function information. The annotation of a constituent consists of both pieces of information joined by a hyphen, e.g., the label NP-TMP is used to annotate a constituent of category NP which functions as a temporal modifier. The syntactic category of a constituent is generally determined by the lexical material in the covered string and the way this material is combined; the syntactic function of a constituent, on the other hand, is determined by the material outside of the constituent. For example, one can determine that the string “last month” can be an NP just based on the string; but we have to look at the surrounding material in order to determine whether “last month” is used as an argument or an adjunct. In consequence, the consistency test of the mapping from strings to their syntactic annotation we are proposing in this paper is most appropriate for the syntactic category annotation, and we thus focus on this annotation for our case study. Nevertheless, the variation n-gram approach is also applicable to syntactic function annotations because a variation n-gram is a nucleus within a context, i.e., within an environment which constrains the syntactic function of the nucleus—something we intend to explore in future work.

For the case study, we used the TIGERRegistry developed at the University of Stuttgart to import the corpus into the TIGER-XML format (König et al., 2003). The TIGERRegistry import filter we used removes the function labels from the categories and places them as edge labels onto the edge above the category; e.g., a temporal noun phrase (NP-TMP) becomes a noun phrase node (NP) under an edge labeled temporal (TMP). The TIGER-XML format allows easy access to the tokens and the syntactic category labels using XSLT, and our variation n-gram algorithm then runs on the result of this process.

Null elements In addition to providing the syntactic category and function annotation, the WSJ annotators also modified the corpus text by inserting so-called null elements, e.g., markers for arguments and adjuncts which are realized non-locally, or unstated units of measurement (cf. Bies et al., 1995, p. 59). The syntactic annotation of these empty elements is largely determined through theoretical considerations and the non-local occurrence of linguistic material, i.e., not the empty element itself or its local context. In other words, the variation in the annotation of a null elements as the nucleus is largely independent of the local environment so that such nuclei should be ignored when testing for the consistency of the mapping from strings to their syntactic annotation. We will see an illustration of this conclusion in the discussion of the case study results in the next section.

Unary branching Finally, we need to discuss a special type of syntactic constituency which is directly relevant when talking about the possible mappings from strings to constituency: categories dominating only a single daughter. The syntactic annotation in the WSJ makes use of such unary branches, which are motivated by theoretical considerations. For our discussion, the important aspect is that a unary branch causes the same string to be annotated by two distinct categories, which would be detected as a variation in the annotation of this string. To instead obtain the interpretation that the two syntactic categories conjunctively characterize the string, we replaced all unary syntactic structures with a category label consisting of the mother and the daughter category. For example, as discussed in Bies et al. (1995), a quantifier phrase (QP) which is missing a head noun, such as *10 million*, in the WSJ is dominated by a noun phrase (NP) node in a unary structure. We replace this structure with the new category label NP/QP. This conversion added 70 syntactic category labels to the original 27 labels[3] used in the WSJ.[4]

4.1 Results from the WSJ

Using the WSJ corpus in the format described above, we ran the variation n-gram method for every possible nucleus size. As shown in Figure 3, the WSJ contains constituents from size 1 to size 271, making these the only possible nucleus sizes.

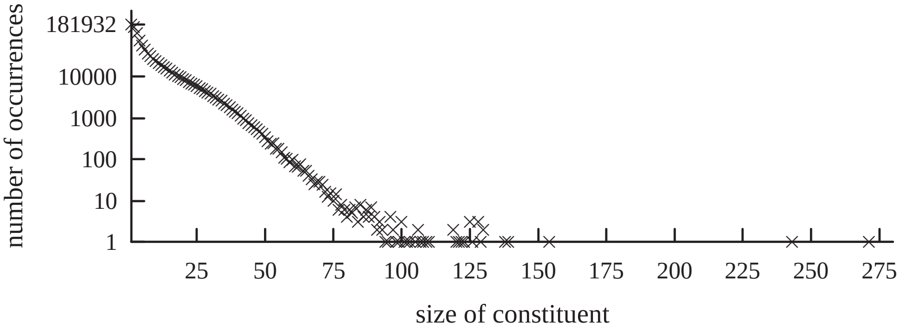

Figure 3: Constituent length in the Wall Street Journal Corpus

The largest repeating string with variation in its annotation is of size 46. Figure 4 shows the number of variation nuclei for sizes 1 through 46.[5] In total, there are 34,564 variation nuclei, more variation than with the part-of-speech variation analysis where Dickinson and Meurers (2003) report 7033 variation nuclei.

[3] The manual (Bies et al., 1995) defines 26 category labels; additionally the part-of-speech label CD occurs as a category label in the corpus, which is probably an error.

[4] Of the 70 added labels, two are noteworthy in that they display multiple levels of unary branching, namely NP/NP/QP for *3 1/2* and PRN/FRAG/WHADJP for *how incompetent at risk assessment and evaluation*. The former appears in variation and is an error, while the latter appears to be correct.

[5] Nucleus sizes with zero counts are omitted.

i	nuclei	i	nuclei	i	nuclei	i	nuclei	i	nuclei
1	9150	8	122	15	8	22	3	35	2
2	14654	9	87	16	10	23	3	36	1
3	6156	10	69	17	10	24	3	37	1
4	2520	11	45	18	9	26	1	45	2
5	1022	12	37	19	9	27	2	46	2
6	393	13	18	20	6	28	4		
7	196	14	15	21	2	31	2	total	34,564

Figure 4: Nucleus size and number of different nuclei

To evaluate the precision of the variation n-gram algorithm, we need to know which of the detected variation nuclei actually include category assignments that are real errors. To do so, we examine the distinct variation nuclei, where by *distinct* we mean that each corpus position is only taken into account for the longest variation n-gram it occurs in.[6] Furthermore, as shown in Dickinson and Meurers (2003), variation nuclei which appear on the fringe of a variation n-gram (i.e., nuclei which border words that are not part of the variation n-gram) are unreliable for determining whether there is an error or not. Thus, for syntactic error detection, we examined only non-fringe nuclei, giving us a total of 6277 distinct variation nuclei, as shown in Figure 5.

i	nuclei	i	nuclei	i	nuclei	i	nuclei	i	nuclei
1	3165	8	92	15	2	22	1	35	3
2	1235	9	75	16	13	23	2	36	2
3	705	10	64	17	10	24	4	37	2
4	338	11	58	18	18	26	5	45	3
5	152	12	37	19	22	27	4	46	4
6	131	13	29	20	6	28	4		
7	82	14	6	21	1	31	2	total	6277

Figure 5: Non-fringe distinct nuclei counts

From these 6277, we randomly sampled 100 examples and marked for each nucleus whether the variation in the annotation of the instances of this nucleus was an annotation error or an ambiguity. We found that 71 out of 100 examples were errors. The 95% confidence interval for the point estimate of .71 is (.6211, .7989), i.e., the number of real errors detected in the 6277 cases is estimated to be between 3898 and 5014. Note that these are counts of distinct variation nuclei

[6] This eliminates the effect that each variation n-gram instance also is an instance of a variation (n-1)-gram (for $n >$ *size-of-nucleus*).

(i.e., recurring strings). By their definition, each variation nucleus has at least two instances which differ in their annotation; each variation that is not an ambiguity thus corresponds to at least one instance (but possibly more instances) of erroneous annotation.

Ambiguities Of the 29 ambiguous nuclei in the sample, ten are a null element (nucleus size of one) that vary between two different categories and the ambiguity arises because the null element occurs in place of an element realized elsewhere. For instance, in example (3), the null element _*EXP*_ (expletive) can be annotated as a sentence (S) or as a relative/subordinate clause (SBAR), depending on the properties of the clause it refers to.

(3) a. For cities losing business to suburban shopping centers , it *EXP*$_S$ may be a wise business investment * [$_S$ to help * keep those jobs and sales taxes within city limits] .

b. But if the market moves quickly enough , it *EXP*$_{SBAR}$ may be impossible [$_{SBAR}$ for the broker to carry out the order] because the investment has passed the specified price .

Removing null elements as variation nuclei of size 1 reduces our set of non-fringe distinct variation nuclei to 5584, and changes our proportion of errors to 71 out of 90 (78.9%). The 95% confidence interval becomes (.7046, .8732), meaning that out of the 5584 examples, we are 95% confident that there are between 3934 and 4875 errors.

Another six ambiguities deal with coordinate structures. In the guidelines of the Penn Treebank (Bies et al., 1995), there is a distinction made for simple and complex coordinate elements. Even if an element is simple (i.e., non-modified), it is annotated like a complex element when it is conjoined with one. In Figure 6, for example, _interest_ is part of a flat structure because all the nouns are simple.

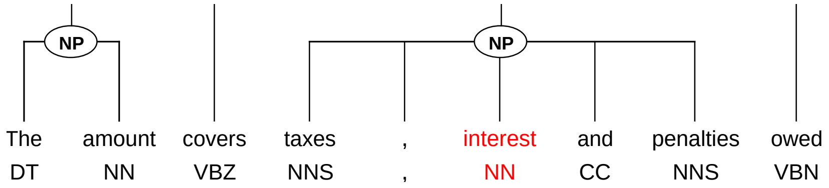

Figure 6: An occurrence of “interest” in a flat structure

In Figure 7, on the other hand, _interest_ is an NP because it conjoins with a modified noun, which must be NP. These coordination ambiguities are systematic, but different from our decision to exclude null elements from being variation nuclei, no simple way to eliminate this ambiguity seems to be available.

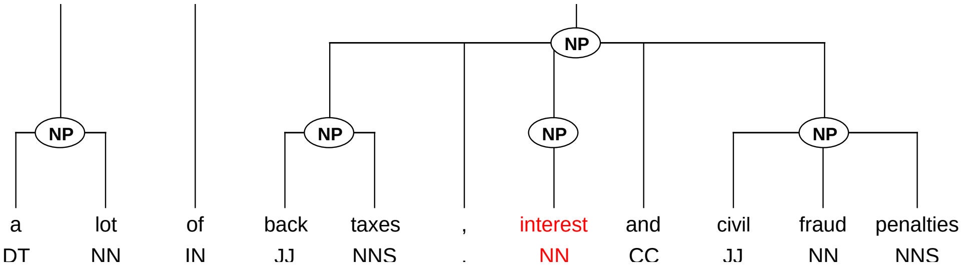

Figure 7: An occurrence of “interest” in a complex coordinate structure

5 Related work

Testing interannotator agreement involves comparing syntactic annotations and thus is related to the variation n-gram approach discussed in this paper. Brants and Skut (1998) discuss automating the comparison of two syntactic annotations of a sentence in interannotator agreement testing. Based on Calder (1997), they propose to select the nodes from one annotation and search for a node with the same terminal yield in the second annotation. The process is driven by the selection of non-terminals from one of the annotations and thus is asymmetric in nature. As a result, the full comparison involves running the process in both directions, selecting from the first annotation and comparing with the second as well as the other way around. In comparison, we have described a symmetric, data-driven method that starts from the occurrence of recurring strings and searches for non-terminals that can cover these strings. The method handles comparisons between more than two sentences since it looks at all occurrences of a given string in parallel. Finally, Brants and Skut (1998) are only concerned with a comparison on the sentence level, whereas the approach we have presented automatically determines comparable strings, which can be smaller or larger than a single sentence.

Wallis (2003) argues for moving from a sentence-by-sentence correction approach (*longitudinal* correction) to what he calls *transverse* correction: correction on a construction-by-construction basis across the whole corpus. The variation n-grams method also looks for consistency across the corpus and thus is an instance of a transverse correction approach; but different from Wallis (2003) who proposes a theory-driven approach in which one manually searches the corpus for particular linguistic constructions, our approach detects variation based on an automatic, data-driven process searching for strings that reoccur in the corpus.

Blaheta (2002) provides an interesting categorization of errors into three classes, based on whether an error is detectable, fixable, or not covered by the annotation guidelines. Human inspection and hand-written rules are used for error detection and correction.

Finally, the variation n-gram method highlights those aspects of an annotation scheme which were difficult for annotators to agree upon. This brings into focus the limit of annotator precision with a given set of guidelines. Sampson and Babarczy (2003) discuss annotation schemes which do not require human annotators to make distinctions that cannot be made reliably and Voutilainen and Järvinen (1995) show that 100% interannotator agreement is possible for both part-of-speech and syntactic annotation when difficult distinctions are eliminated.

6 Summary

We have shown how an approach to treebank error detection based on so-called variation n-grams can be defined and illustrated with a case study based on the WSJ treebank that it successfully detects inconsistencies in syntactic category annotation. Since such inconsistencies are generally introduced by humans our method works best for large corpora that have been annotated manually or semi-automatically, which is generally the case for current syntactic and other high-level annotation.

Our work serves two main purposes for treebank improvement. It is a means for finding erroneous variation in a corpus, which can then be corrected. And it provides feedback for the development of empirically adequate standards for syntactic annotation, showing which distinctions are difficult to maintain over an entire corpus. Additionally, as a method for comparing syntactic annotation, our work could have uses for interannotator agreement testing and parser evaluation.

References

Abeillé, Anne (ed.) (2003). *Treebanks: Building and using syntactically annotated corpora*. Dordrecht: Kluwer. http://treebank.linguist.jussieu.fr/toc.html.

Agrawal, Rakesh and Ramakrishnan Srikant (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases*. Morgan Kaufmann, pp. 487–499. http://www.almaden.ibm.com/cs/people/ragrawal/papers/vldb94.ps.

Bies, Ann, Mark Ferguson, Karen Katz and Robert MacIntyre (1995). *Bracketing*

Guidelines for Treebank II Style Penn Treebank Project. University of Pennsylvania. ftp://ftp.cis.upenn.edu/pub/treebank/doc/manual/root.ps.gz.

Blaheta, Don (2002). Handling noisy training and testing data. In *Proceedings of the 7th conference on Empirical Methods in Natural Language Processing*. pp. 111–116. http://www.cs.brown.edu/~dpb/papers/dpb-emnlp02.html.

Brants, Thorsten and Wojciech Skut (1998). Automation of Treebank Annotation. In *Proceedings of New Methods in Language Processing*. Syndey, Australia. http://www.coli.uni-sb.de/~thorsten/publications/Brants-Skut-NeMLaP98.ps.gz

Calder, Jo (1997). On aligning trees. In *Proceedings of the Second Conference of Empirical Methods in Natural Language Processing*. Brown University. http://xxx.lanl.gov/abs/cmp-lg/9707016.

Dickinson, Markus and W. Detmar Meurers (2003). 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. http://ling.osu.edu/~dm/papers/dickinson-meurers-03.html.

Eskin, Eleazar (2000). Automatic Corpus Correction with Anomaly Detection. In *Proceedings of the First Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-00)*. Seattle, Washington. http://www.cs.columbia.edu/~eeskin/papers/treebank-anomaly-naacl00.ps.

König, Esther, Wolfgang Lezius and Holger Voormann (2003). *TIGERSearch User's Manual*. IMS, University of Stuttgart. http://www.tigersearch.de.

Marcus, Mitchell, Beatrice Santorini, Mary Ann Marcinkiewicz and Ann Taylor (1999). Treebank-3 Corpus. Linguistic Data Consortium. Philadelphia. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC99T42.

Sampson, Geoffrey and Anna Babarczy (2003). Limits to annotation precision. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*. pp. 61–68. http://www.grsampson.net/Alta.html.

Taylor, Ann, Mitchell Marcus and Beatrice Santorini (2003). The Penn Treebank: An Overview. In Abeillé (2003), pp. 5–22. http://treebank.linguist.jussieu.fr/pdf/1.pdf.

van Halteren, Hans (2000). The Detection of Inconsistency in Manually Tagged Text. In Anne Abeillé, Thosten Brants and Hans Uszkoreit (eds.), *Proceedings of the Second Workshop on Linguistically Interpreted Corpora*. Luxembourg.

Voutilainen, Atro and Timo Järvinen (1995). Specifying a shallow grammatical representation for parsing purposes. In *Proceedings of the 7th Conference of the EACL*. Dublin, Ireland. http://www.aclweb.org/anthology/E95-1029.

Wallis, Sean (2003). Completing Parsed Corpora. In Abeillé (2003), pp. 51–71. http://treebank.linguist.jussieu.fr/pdf/4.pdf.