Detecting Errors in Discontinuous Structural Annotation

Markus Dickinson
Department of Linguistics
The Ohio State University
dickinso@ling.osu.edu

W. Detmar Meurers
Department of Linguistics
The Ohio State University
dm@ling.osu.edu

Abstract

Consistency of corpus annotation is an essential property for the many uses of annotated corpora in computational and theoretical linguistics. While some research addresses the detection of inconsistencies in positional annotation (e.g., part-of-speech) and continuous structural annotation (e.g., syntactic constituency), no approach has yet been developed for automatically detecting annotation errors in discontinuous structural annotation. This is significant since the annotation of potentially discontinuous stretches of material is increasingly relevant, from treebanks for free-word order languages to semantic and discourse annotation.

In this paper we discuss how the variation n-gram error detection approach (Dickinson and Meurers, 2003a) can be extended to discontinuous structural annotation. We exemplify the approach by showing how it successfully detects errors in the syntactic annotation of the German TIGER corpus (Brants et al., 2002).

1 Introduction

Annotated corpora have at least two kinds of uses: firstly, as training material and as "gold standard" testing material for the development of tools in computational linguistics, and secondly, as a source of data for theoretical linguists searching for analytically relevant language patterns.

Annotation errors and why they are a problem The high quality annotation present in "gold standard" corpora is generally the result of a manual or semi-automatic mark-up process. The annotation thus can contain annotation errors from automatic (pre-)processes, human post-editing, or human annotation. The presence of errors creates problems for both computational and theoretical linguistic uses, from unreliable training and evaluation of natural language processing technology (e.g., van Halteren, 2000; Květoň and Oliva, 2002, and the work mentioned below) to low precision and recall of queries for already rare linguistic phenomena. Investigating the quality of linguistic annotation and improving it where possible thus is a key issue for the use of annotated corpora in computational and theoretical linguistics.

Illustrating the negative impact of annotation errors on computational uses of annotated corpora, van Halteren et al. (2001) compare taggers trained and tested on the Wall Street Journal (WSJ, Marcus et al., 1993) and the Lancaster-Oslo-Bergen (LOB, Johansson, 1986) corpora and find that the results for the WSJ perform significantly worse. They report that the lower accuracy figures are caused by inconsistencies in the WSJ annotation and that 44% of the errors for their best tagging system were caused by "inconsistently handled cases."

Turning from training to evaluation, Padro and Marquez (1998) highlight the fact that the true accuracy of a classifier could be much better or worse than reported, depending on the error rate of the corpus used for the evaluation. Evaluating two taggers on the WSJ, they find tagging accuracy rates for am-

Proceedings of the 43rd Annual Meeting of the ACL, pages 322–329,
Ann Arbor, June 2005.

biguous words of 91.35% and 92.82%. Given the estimated 3% error rate of the WSJ tagging (Marcus et al., 1993), they argue that the difference in performance is not sufficient to establish which of the two taggers is actually better.

In sum, corpus annotation errors, especially errors which are inconsistencies, can have a profound impact on the quality of the trained classifiers and the evaluation of their performance. The problem is compounded for syntactic annotation, given the difficulty of evaluating and comparing syntactic structure assignments, as known from the literature on parser evaluation (e.g., Carroll et al., 2002).

The idea that variation in annotation can indicate annotation errors has been explored to detect errors in part-of-speech (POS) annotation (van Halteren, 2000; Eskin, 2000; Dickinson and Meurers, 2003a) and syntactic annotation (Dickinson and Meurers, 2003b). But, as far as we are aware, the research we report on here is the first approach to error detection for the increasing number of annotations which make use of more general graph structures for the syntactic annotation of free word order languages or the annotation of semantic and discourse properties.

Discontinuous annotation and its relevance The simplest kind of annotation is positional in nature, such as the association of a part-of-speech tag with each corpus position. On the other hand, structural annotation such as that used in syntactic treebanks (e.g., Marcus et al., 1993) assigns a syntactic category to a contiguous sequence of corpus positions. For languages with relatively free constituent order, such as German, Dutch, or the Slavic languages, the combinatorial potential of the language encoded in constituency cannot be mapped straightforwardly onto the word order possibilities of those languages. As a consequence, the treebanks that have been created for German (NEGRA, Skut et al., 1997; VERBMOBIL, Hinrichs et al., 2000; TIGER, Brants et al., 2002) have relaxed the requirement that constituents have to be contiguous. This makes it possible to syntactically annotate the language data as such, i.e., without requiring postulation of empty elements as placeholders or other theoretically motivated changes to the data. We note in passing that discontinuous constituents have also received some support in theoretical linguistics (cf., e.g., the articles collected in Huck and Ojeda, 1987; Bunt and van Horck, 1996).

Discontinuous constituents are strings of words which are not necessarily contiguous, yet form a single constituent with a single label, such as the noun phrase *Ein Mann, der lacht* in the German relative clause extraposition example (1) (Brants et al., 2002).[1]

(1) **Ein** **Mann** kommt , **der** **lacht**
a man comes , who laughs
'A man who laughs comes.'

In addition to their use in syntactic annotation, discontinuous structural annotation is also relevant for semantic and discourse-level annotation—essentially any time that graph structures are needed to encode relations that go beyond ordinary tree structures. Such annotations are currently employed in the mark-up for semantic roles (e.g., Kingsbury et al., 2002) and multi-word expressions (e.g., Rayson et al., 2004), as well as for spoken language corpora or corpora with multiple layers of annotation which cross boundaries (e.g., Blache and Hirst, 2000).

In this paper, we present an approach to the detection of errors in discontinuous structural annotation. We focus on syntactic annotation with potentially discontinuous constituents and show that the approach successfully deals with the discontinuous syntactic annotation found in the TIGER treebank (Brants et al., 2002).

2 The variation n-gram method

Our approach builds on the variation n-gram algorithm introduced in Dickinson and Meurers (2003a,b). The basic idea behind that approach is that a string occurring more than once can occur with different labels in a corpus, which we refer to as *variation*. Variation is caused by one of two reasons: i) *ambiguity*: there is a type of string with multiple possible labels and different corpus occurrences of that string realize the different options, or ii) *error*: the tagging of a string is inconsistent across comparable occurrences.

[1] The ordinary way of marking a constituent with brackets is inadequate for discontinuous constituents, so we instead boldface and underline the words belonging to a discontinuous constituent.

The more similar the context of a variation, the more likely the variation is an error. In Dickinson and Meurers (2003a), contexts are composed of words, and identity of the context is required. The term *variation n-gram* refers to an n-gram (of words) in a corpus that contains a string annotated differently in another occurrence of the same n-gram in the corpus. The string exhibiting the variation is referred to as the *variation nucleus*.

2.1 Detecting variation in POS annotation

In Dickinson and Meurers (2003a), we explore this idea for part-of-speech annotation. For example, in the WSJ corpus the string in (2) is a variation 12-gram since *off* is a variation nucleus that in one corpus occurrence is tagged as a preposition (IN), while in another it is tagged as a particle (RP).[2]

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

Once the variation n-grams for a corpus have been computed, heuristics are employed to classify the variations into errors and ambiguities. The first heuristic encodes the basic fact that the label assignment for a nucleus is dependent on the context: variation nuclei in long n-grams are likely to be errors. The second takes into account that natural languages favor the use of local dependencies over non-local ones: nuclei found at the fringe of an n-gram are more likely to be genuine ambiguities than those occurring with at least one word of surrounding context. Both of these heuristics are independent of a specific corpus, annotation scheme, or language.

We tested the variation error detection method on the WSJ and found 2495 distinct[3] nuclei for the variation n-grams between the 6-grams and the 224-grams. 2436 of these were actual errors, making for a precision of 97.6%, which demonstrates the value of the long context heuristic. 57 of the 59 genuine ambiguities were fringe elements, confirming that fringe elements are more indicative of a true ambiguity.

[2]To graphically distinguish the variation nucleus within a variation n-gram, the nucleus is shown in grey.

[3]Being distinct means that each corpus position is only taken into account for the longest variation n-gram it occurs in.

2.2 Detecting variation in syntactic annotation

In Dickinson and Meurers (2003b), we decompose the variation n-gram detection for syntactic annotation into a series of runs with different nucleus sizes. This is needed to establish a one-to-one relation between a unit of data and a syntactic category annotation for comparison. 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, the approach ensures that all strings which are analyzed as a constituent somewhere in the corpus are compared to the annotation of all other occurrences of that string.

For example, the variation 4-gram *from a year earlier* appears 76 times in the WSJ, where the nucleus *a year* is labeled noun phrase (NP) 68 times, and 8 times it is not annotated as a constituent and is given the special label NIL. 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 is labeled as a noun phrase (NP) and once as a prepositional phrase (PP).

To be able to efficiently calculate all variation nuclei of a treebank, in Dickinson and Meurers (2003b) we make use of the fact that a variation necessarily involves at least one constituent occurrence of a nucleus and calculate the set of nuclei for a window of length i by first finding the constituents of that length. Based on this set, we then find non-constituent occurrences of all strings occurring as constituents. Finally, the variation n-grams for these variation nuclei are obtained in the same way as for POS annotation.

In the WSJ, the method found 34,564 variation nuclei, up to size 46; an estimated 71% of the 6277 non-fringe distinct variation nuclei are errors.

3 Discontinuous constituents

In Dickinson and Meurers (2003b), we argued that null elements need to be ignored as variation nuclei because the variation in the annotation of a null element as the nucleus is largely independent of the local environment. For example, in (3) the null element **EXP** (expletive) can be annotated a. as a sentence (S) or b. 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* 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* may be impossible [$_{SBAR}$ for the broker to carry out the order] because the investment has passed the specified price .

We found that removing null elements as variation nuclei of size 1 increased the precision of error detection to 78.9%.

Essentially, null elements represent discontinuous constituents in a formalism with a context-free backbone (Bies et al., 1995). Null elements are coindexed with a non-adjacent constituent; in the predicate argument structure, the constituent should be interpreted where the null element is.

To be able to annotate discontinuous material without making use of inserted null elements, some treebanks have instead relaxed the definition of a linguistic tree and have developed more complex graph annotations. An error detection method for such corpora thus does not have to deal with the problems arising from inserted null elements discussed above, but instead it must function appropriately even if constituents are discontinuously realized.

A technique such as the variation n-gram method is applicable to corpora with a one-to-one mapping between the text and the annotation. For corpora with positional annotation—e.g., part-of-speech annotated corpora—the mapping is trivial given that the annotation consists of one-to-one correspondences between words (i.e., tokens) and labels. For corpora annotated with more complex structural information—e.g., syntactically-annotated corpora—the one-to-one mapping is obtained by considering every interval (continuous string of any length) which is assigned a category label somewhere in the corpus.

While this works for treebanks with continuous constituents, a one-to-one mapping is more complicated to establish for syntactic annotation involving discontinuous constituents (NEGRA, Skut et al., 1997; TIGER, Brants et al., 2002). In order to apply the variation n-gram method to discontinuous constituents, we need to develop a technique which is capable of comparing labels for any set of corpus positions, instead of for any interval.

4 Extending the variation n-gram method

To extend the variation n-gram method to handle discontinuous constituents, we first have to define the characteristics of such a constituent (section 4.1), in other words our units of data for comparison. Then, we can find identical non-constituent (NIL) strings (section 4.2) and expand the context into variation n-grams (section 4.3).

4.1 Variation nuclei: Constituents

For traditional syntactic annotation, a variation nucleus is defined as a contiguous string with a single label; this allows the variation n-gram method to be broken down into separate runs, one for each constituent size in the corpus. For discontinuous syntactic annotation, since we are still interested in comparing cases where the nucleus is the same, we will treat two constituents as having the same size if they consist of the same number of words, regardless of the amount of intervening material, and we can again break the method down into runs of different sizes. The intervening material is accounted for when expanding the context into n-grams.

A question arises concerning the word order of elements in a constituent. Consider the German example (4) (Müller, 2004).

(4) weil der Mann$_{nom}$ der Frau$_{dat}$ das Buch$_{acc}$ gab.
because the man the woman the book gave
'because the man gave the woman the book.'

The three arguments of the verb *gab* ('give') can be permuted in all six possible ways and still result in a well-formed sentence. It might seem, then, that we would want to allow different permutations of nuclei to be treated as identical. If *das Buch der Frau gab* is a constituent in another sentence, for instance, it should have the same category label as *der Frau das Buch gab*.

Putting all permutations into one equivalence class, however, amounts to stating that all order-

ings are always the same. But even "free word order" languages are more appropriately called free constituent order; for example, in (4), the argument noun phrases can be freely ordered, but each argument noun phrase is an atomic unit, and in each unit the determiner precedes the noun.

Since we want our method to remain data-driven and order can convey information which might be reflected in an annotation system, we keep strings with different orders of the same words distinct, i.e., ordering of elements is preserved in our method.

4.2 Variation nuclei: Non-constituents

The basic idea is to compare a string annotated as a constituent with the same string found elsewhere—whether annotated as a constituent or not. So we need to develop a method for finding all string occurrences not analyzed as a constituent (and assign them the special category label NIL). Following Dickinson and Meurers (2003b), we only look for non-constituent occurrences of those strings which also occur at least once as a constituent.

But do we need to look for discontinuous NIL strings or is it sufficient to assume only continuous ones? Consider the TIGER treebank examples (5).

(5) a. in diesem Punkt seien **sich** Bonn und
on this point are SELF Bonn and
London nicht **einig** .
London not agreed .
'Bonn and London do not agree on this point.'

b. in diesem Punkt seien **sich** Bonn und
on this point are SELF Bonn and
London offensichtlich **nicht einig** .
London clearly not agreed .

In example (5a), *sich einig* ('SELF agree') forms an adjective phrase (AP) constituent. But in example (5b), that same string is not analyzed as a constituent, despite being in a nearly identical sentence. We would thus like to assign the discontinuous string *sich einig* in (5b) the label NIL, so that the labeling of this string in (5a) can be compared to its occurrence in (5b).

In consequence, our approach should be able to detect NIL strings which are discontinuous—an issue which requires special attention to obtain an algorithm efficient enough to handle large corpora.

Use sentence boundary information The first consideration makes use of the fact that syntactic annotation by its nature respects sentence boundaries. In consequence, we never need to search for NIL strings that span across sentences.[4]

Use tries to store constituent strings The second consideration concerns how we calculate the NIL strings. To find every non-constituent string in the corpus, discontinuous or not, which is identical to some constituent in the corpus, a basic approach would first generate all possible strings within a sentence and then test to see which ones occur as a constituent elsewhere in the corpus. For example, if the sentence is *Nobody died when Clinton lied*, we would see if any of the 31 subsets of strings occur as constituents (e.g., *Nobody*, *Nobody when*, *Clinton lied*, *Nobody when lied*, etc.). But such a generate and test approach clearly is intractable given that it generates generates $2^n - 1$ potential matches for a sentence of n words.

We instead split the task of finding NIL strings into two runs through the corpus. In the first, we store all constituents in the corpus in a trie data structure (Fredkin, 1960), with words as nodes. In the second run through the corpus, we attempt to match the strings in the corpus with a path in the trie, thus identifying all strings occurring as constituents somewhere in the corpus.

Filter out unwanted NIL strings The final consideration removes "noisy" NIL strings from the candidate set. Certain NIL strings are known to be useless for detecting annotation errors, so we should remove them to speed up the variation n-gram calculations. Consider example (6) from the TIGER corpus, where the continuous constituent *die Menschen* is annotated as a noun phrase (NP).

(6) Ohne diese Ausgaben, so die
without these expenses according to the
Weltbank, seien **die Menschen** totes Kapital
world bank are the people dead capital
'According to the world bank, the people are dead capital without these expenses.'

[4] This restriction clearly is syntax specific and other topological domains need to be identified to make searching for NIL strings tractable for other types of discontinuous annotation.

Our basic method of finding NIL strings would detect another occurrence of *die Menschen* in the same sentence since nothing rules out that the other occurrence of *die* in the sentence (preceding *Weltbank*) forms a discontinuous NIL string with *Menschen*. Comparing a constituent with a NIL string that contains one of the words of the constituent clearly goes against the original motivation for wanting to find discontinuous strings, namely that they show variation between different occurrences of a string.

To prevent such unwanted variation, we eliminate occurrences of NIL-labeled strings that overlap with identical constituent strings from consideration.

4.3 Variation n-grams

The more similar the context surrounding a variation nucleus, the more likely it is for a variation in its annotation to be an error. For detecting errors in traditional syntactic annotation (see section 2.2), the context consists of the elements to the left and the right of the nucleus. When nuclei can be discontinuous, however, there can also be *internal context*, i.e., elements which appear between the words forming a discontinuous variation nucleus.

As in our earlier work, an instance of the a priori algorithm is used to expand a nucleus into a longer n-gram by stepwise adding context elements. Where previously it was possible to add an element to the left or the right, we now also have the option of adding it in the middle—as part of the new, internal context. But depending on how we fill in the internal context, we can face a serious tractability problem. Given a nucleus with j gaps within it, we need to potentially expand it in $j + 2$ directions, instead of in just 2 directions (to the right and to the left).

For example, the potential nucleus *was werden* appears as a verb phrase (VP) in the TIGER corpus in the string *was ein Seeufer werden*; elsewhere in the corpus *was* and *werden* appear in the same sentence with 32 words between them. The chances of one of the middle 32 elements matching something in the internal context of the VP is relatively high, and indeed the twenty-sixth word is *ein*. However, if we move stepwise out from the nucleus in order to try to match *was ein Seeufer werden*, the only options are to find *ein* directly to the right of *was* or *Seeufer* directly to the left of *werden*, neither of which occurs, thus stopping the search.

In conclusion, we obtain an efficient application of the a priori algorithm by expanding the context only to elements which are adjacent to an element already in the n-gram. Note that this was already implicitly assumed for the left and the right context.

There are two other efficiency-related issues worth mentioning. Firstly, as with the variation nucleus detection, we limit the n-grams expansion to sentences only. Since the category labels do not represent cross-sentence dependencies, we gain no new information if we find more context outside the sentence, and in terms of efficiency, we cut off what could potentially be a very large search space.[5]

Secondly, the methods for reducing the number of variation nuclei discussed in section 4.2 have the consequence of also reducing the number of possible variation n-grams. For example, in a test run on the NEGRA corpus we allowed identical strings to overlap; this generated a variation nucleus of size 63, with 16 gaps in it, varying between NP and NIL within the same sentence. Fifteen of the gaps can be filled in and still result in variation. The filter for unwanted NIL strings described in the previous section eliminates the NIL value from consideration. Thus, there is no variation and no tractability problem in constructing n-grams.

4.3.1 Generalizing the n-gram context

So far, we assumed that the context added around variation nuclei consists of words. Given that treebanks generally also provide part-of-speech information for every token, we experimented with part-of-speech tags as a less restrictive kind of context. The idea is that it should be possible to find more variation nuclei with comparable contexts if only the part-of-speech tags of the surrounding words have to be identical instead of the words themselves.

As we will see in section 5, generalizing n-gram contexts in this way indeed results in more variation n-grams being found, i.e., increased recall.

4.4 Adapting the heuristics

To determine which nuclei are errors, we can build on the two heuristics from previous research (Dick-

[5] Note that similar sentences which were segmented differently could potentially cause varying n-gram strings not to be found. We propose to treat this as a separate sentence segmentation error detection phase in future work.

inson and Meurers, 2003a,b)—trust long contexts and distrust the fringe—with some modification, given that we have more fringe areas to deal with for discontinuous strings. In addition to the right and the left fringe, we also need to take into account the internal context in a way that maintains the non-fringe heuristic as a good indicator for errors. As a solution that keeps internal context on a par with the way external context is treated in our previous work, we require one word of context around every terminal element that is part of the variation nucleus. As discussed below, this heuristic turns out to be a good predictor of which variations are annotation errors; expanding to the longest possible context, as in Dickinson and Meurers (2003a), is not necessary.

5 Results on the TIGER Corpus

We ran the variation n-grams error detection method for discontinuous syntactic constituents on v. 1 of TIGER (Brants et al., 2002), a corpus of 712,332 tokens in 40,020 sentences. The method detected a total of 10,964 variation nuclei. From these we sampled 100 to get an estimate of the number of errors in the corpus which concern variation. Of these 100, 13 variation nuclei pointed to an error; with this point estimate of .13, we can derive a 95% confidence interval of (0.0641, 0.1959),[6] which means that we are 95% confident that the true number of variation-based errors is between 702 and 2148. The effectiveness of a method which uses context to narrow down the set of variation nuclei can be judged by how many of these variation errors it finds.

Using the non-fringe heuristic discussed in the previous section, we selected the shortest non-fringe variation n-grams to examine. Occurrences of the same strings within larger n-grams were ignored, so as not to artificially increase the resulting set of n-grams.

When the context is defined as identical words, we obtain 500 variation n-grams. Sampling 100 of these and labeling for each position whether it is an error or an ambiguity, we find that 80 out of the 100 samples point to at least one token error. The 95% confidence interval for this point estimate of .80 is (0.7216, 0.8784), so we are 95% confident that the true number of error types is between 361 and 439. Note that this precision is comparable to the estimates for continuous syntactic annotation in Dickinson and Meurers (2003b) of 71% (with null elements) and 78.9% (without null elements).

When the context is defined as identical parts of speech, as described in section 4.3.1, we obtain 1498 variation n-grams. Again sampling 100 of these, we find that 52 out of the 100 point to an error. And the 95% confidence interval for this point estimate of .52 is (0.4221, 0.6179), giving a larger estimated number of errors, between 632 and 926.

Context	Precision	Errors
Word	80%	361–439
POS	52%	632–926

Figure 1: Accuracy rates for the different contexts

Words convey more information than part-of-speech tags, and so we see a drop in precision when using part-of-speech tags for context, but these results highlight a very practical benefit of using a generalized context. By generalizing the context, we maintain a precision rate of approximately 50%, and we substantially increase the recall of the method. There are, in fact, likely twice as many errors when using POS contexts as opposed to word contexts. Corpus annotation projects willing to put in some extra effort thus can use this method of finding variation n-grams with a generalized context to detect and correct more errors.

6 Summary and Outlook

We have described the first method for finding errors in corpora with graph annotations. We showed how the variation n-gram method can be extended to discontinuous structural annotation, and how this can be done efficiently and with as high a precision as reported for continuous syntactic annotation. Our experiments with the TIGER corpus show that generalizing the context to part-of-speech tags increases recall while keeping precision above 50%. The method can thus have a substantial practical benefit when preparing a corpus with discontinuous annotation.

Extending the error detection method to handle

[6] The 95% confidence interval was calculated using the standard formula of $p \pm 1.96\sqrt{\frac{p(1-p)}{n}}$, where p is the point estimate and n the sample size.

discontinuous constituents, as we have done, has significant potential for future work given the increasing number of free word order languages for which corpora and treebanks are being developed.

Acknowledgements We are grateful to George Smith and Robert Langner of the University of Potsdam TIGER team for evaluating the variation we detected in the samples. We would also like to thank the three ACL reviewers for their detailed and helpful comments, and the participants of the OSU CLippers meetings for their encouraging feedback.

References

Ann Bies, Mark Ferguson, Karen Katz and Robert MacIntyre, 1995. *Bracketing Guidelines for Treebank II Style Penn Treebank Project*. University of Pennsylvania.

Philippe Blache and Daniel Hirst, 2000. Multi-level annotation for spoken-language corpora. In *Proceedings of ICSLP-00*. Beijing, China.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius and George Smith, 2002. The TIGER Treebank. In *Proceedings of TLT-02*. Sozopol, Bulgaria.

Harry Bunt and Arthur van Horck (eds.), 1996. *Discontinuous Constituency*. Mouton de Gruyter, Berlin and New York.

John Carroll, Anette Frank, Dekang Lin, Detlef Prescher and Hans Uszkoreit (eds.), 2002. *Proceedings of the LREC Workshop "Beyond PARSEVAL. Towards Improved Evaluation Measures for Parsing Systems"*, Las Palmas, Gran Canaria.

Markus Dickinson and W. Detmar Meurers, 2003a. Detecting Errors in Part-of-Speech Annotation. In *Proceedings of EACL-03*. Budapest, Hungary.

Markus Dickinson and W. Detmar Meurers, 2003b. Detecting Inconsistencies in Treebanks. In *Proceedings of TLT-03*. Växjö, Sweden.

Eleazar Eskin, 2000. Automatic Corpus Correction with Anomaly Detection. In *Proceedings of NAACL-00*. Seattle, Washington.

Edward Fredkin, 1960. Trie Memory. *CACM*, 3(9):490–499.

Erhard Hinrichs, Julia Bartels, Yasuhiro Kawata, Valia Kordoni and Heike Telljohann, 2000. The Tübingen Treebanks for Spoken German, English, and Japanese. In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer, Berlin, pp. 552–576.

Geoffrey Huck and Almerindo Ojeda (eds.), 1987. *Discontinuous Constituency*. Academic Press, New York.

Stig Johansson, 1986. *The Tagged LOB Corpus: Users' Manual*. Norwegian Computing Centre for the Humanities, Bergen.

Paul Kingsbury, Martha Palmer and Mitch Marcus, 2002. Adding Semantic Annotation to the Penn TreeBank. In *Proceedings of HLT-02*. San Diego.

Pavel Květŏn and Karel Oliva, 2002. Achieving an Almost Correct PoS-Tagged Corpus. In Petr Sojka, Ivan Kopeček and Karel Pala (eds.), *TSD 2002*. Springer, Heidelberg, pp. 19–26.

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

Stefan Müller, 2004. Continuous or Discontinuous Constituents? A Comparison between Syntactic Analyses for Constituent Order and Their Processing Systems. *Research on Language and Computation*, 2(2):209–257.

Lluis Padro and Lluis Marquez, 1998. On the Evaluation and Comparison of Taggers: the Effect of Noise in Testing Corpora. In *COLING/ACL-98*.

Paul Rayson, Dawn Archer, Scott Piao and Tony McEnery, 2004. The UCREL Semantic Analysis System. In *Proceedings of the Workshop on Beyond Named Entity Recognition: Semantic labelling for NLP tasks*. Lisbon, Portugal, pp. 7–12.

Wojciech Skut, Brigitte Krenn, Thorsten Brants and Hans Uszkoreit, 1997. An Annotation Scheme for Free Word Order Languages. In *Proceedings of ANLP-97*. Washington, D.C.

Hans van Halteren, 2000. The Detection of Inconsistency in Manually Tagged Text. In Anne Abeillé, Thorsten Brants and Hans Uszkoreit (eds.), *Proceedings of LINC-00*. Luxembourg.

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