In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03). Budapest, Hungary.

Detecting Errors in Part-of-Speech 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

We propose a new method for detecting errors in "gold-standard" part-of-speech annotation. The approach locates errors with high precision based on n-grams occurring in the corpus with multiple taggings. Two further techniques, closed-class analysis and finite-state tagging guide patterns, are discussed. The success of the three approaches is illustrated for the Wall Street Journal corpus as part of the Penn Treebank.

1 Introduction

Part-of-speech (pos) annotated reference corpora, such as the British National Corpus (Leech et al., 1994), the Penn Treebank (Marcus et al., 1993), or the German Negra Treebank (Skut et al., 1997) play an important role for current work in computational linguistics. They provide training material for research on tagging algorithms and they serve as a gold standard for evaluating the performance of such tools. High quality, pos-annotated text is also relevant as input for syntactic processing, for practical applications such as information extraction, and for linguistic research making use of pos-based corpus queries.

The gold-standard pos-annotation for such large reference corpora is generally obtained using an automatic tagger to produce a first annotation, followed by human post-editing. While Sinclair (1992) provides some arguments for prioritizing a fully automated analysis, human post-editing has been shown to significantly reduce the number of pos-annotation errors. Brants (2000) discusses that a single human post-editor reduces the 3.3% error rate in the STTS annotation of the German Negra corpus produced by the TnT tagger to 1.2%. Baker (1997) also reports an improvement of around 2% for a similar experiment carried out for an English sample originally tagged with 96.95% accuracy by the CLAWS tagger. And Leech (1997) reports that manual post-editing and correction done for the 2-million word core corpus portion of the BNC, the BNC-sampler, reduced the approximate error rate of 1.7% for the automatically obtained annotation to less than 0.3%.

While the last figure clearly is a remarkable result, van Halteren (2000), working with the written half of the BNC-sampler, reports that in 13.6% of the cases where his WPDV tagger disagrees with the BNC annotation, the cause is an error in the BNC annotation.[1] Improving the correctness of such gold-standard annotation thus is important for obtaining reliable testing material for pos-tagger research, as well as for the other uses of gold-standard annotation mentioned at the beginning of this section—a point which becomes even stronger when one considers that the pos-annotation of most reference corpora contain significantly more errors than the 0.3% figure reported for the BNC-sampler.

In this paper, we present three methods for automatic detection of annotation errors which remain

[1] The percentage of disagreement caused by BNC errors rises to 20.5% for a tagger trained on the entire corpus.

despite human post-editing, and sometimes are actually caused by it. Our main proposal discussed in section 2.1 is independent of the language and tagset of the corpus and requires no additional language resources such as lexica. It detects variation in the pos-annotation of a corpus by searching for n-grams which occur more than once in the corpus and include at least one difference in their annotation. We discuss how all such *variation n-grams* of a corpus can be obtained and show that together with some heuristics they are highly accurate predictors of annotation errors. In section 2.2 we turn to two other simple ideas for detecting pos-annotation errors, *closed-class analysis* and *finite-state tagging guide patterns*. Finally, in section 3 we relate our research to several recent publications addressing the topic of pos-error correction.

2 Three methods for detecting errors

The task of correcting part-of-speech annotation can be viewed as consisting of two steps: i) detecting which corpus positions are incorrectly tagged, and ii) finding the correct tag for those positions.

The first step, *detection*, considers each corpus position and classifies the tag of that position as correct or incorrect. Given that this task involves each corpus position, only a fully automatic detection method is feasible for a large corpus.

The second step, *repair*, considers those positions marked as errors and determines the correct tag. Taking the performance of current automatic taggers as baseline for the quality of the "gold-standard" pos-annotation we intend to correct, for English we can assume that repair needs to consider less than 3% of the number of corpus positions. This makes automation of this second step less critical, as long as the error detection step has a high precision (which is relevant since the repair step also needs to deal with false positives from detection).

Our research in this paper addresses the first issue, detecting errors, and based on the just mentioned reasoning we focus on detecting errors automatically and with high precision.[2] To do so, we propose three different methods, the first relying on internal corpus variation, the second on closed lexical classes, and the third on patterns in the tagging guide. We illustrate the applicability and effectiveness of each method by reporting the results of applying them to the Wall Street Journal (WSJ) corpus as part of the Penn Treebank 3 release, which was tagged using the PARTS tagger and manually corrected afterwards (Marcus et al., 1993).

2.1 Using the variation in a corpus

For each word that occurs in a corpus, there is a lexically determined set of tags that can in principle be assigned to this word. 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. We will refer to this as *variation*.

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,[3] or ii) *error*: the tagging of a word is inconsistent across comparable occurrences. We can therefore locate annotation errors by zooming in on the variation exhibited by a corpus, provided we have a way to decide whether a particular variation is an ambiguity or an error—but how can this be done?

2.1.1 Variation n-grams

The key to answering the question lies in a classification of contexts: the more similar the context of a variation, the more likely it is for the variation to be an error. But we need to make concrete what kind of properties the context consists of and what counts as similar contexts. In this paper, we focus on contexts composed of words[4] and we require identity of the context, not just similarity. We will use the term *variation n-gram* for an

[2] Recall is less relevant in our context since eliminating any substantial number of errors from a "gold-standard" is a worthwhile enterprise. In section 3 we discuss other approaches, which can be combined with ours to raise recall.

[3] 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*.

[4] Other options allowing for application to more corpus instances would be to use contexts composed of pos-tags or some other syntactic or morphological properties.

n-gram (of words) in a corpus that contains a word that is annotated differently in another occurrence of the same n-gram in the corpus. The word exhibiting the variation is referred to as the *variation nucleus*.

For example, in the WSJ, 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

Note that the variation 12-gram in (1) contains two variation 11-grams, which one obtains by eliminating either the first or the last word.

Algorithm To compute all variation n-grams of a corpus, we make use of the just mentioned fact that a variation n-gram must contain a variation $(n-1)$-gram to obtain an algorithm efficient enough to handle large corpora. The algorithm, which essentially is an instance of the a priori algorithm used in information extraction (Agrawal and Srikant, 1994), takes a pos-annotated corpus and outputs a listing of the variation n-grams, from $n = 1$ to the longest n for which there is a variation n-gram in the corpus.

1. Calculate the set of variation unigrams in the corpus and store the variation unigrams and their corpus positions.

2. Based on the corpus positions of the variation n-grams last stored, extend the n-grams to either side (unless the corpus ends there). For each resulting $(n+1)$-gram, check whether it has another instance in the corpus and if there is variation in the way the different occurrences of the $(n+1)$-gram are tagged. Store all variation $(n+1)$-grams and their corpus positions.

3. Repeat step 2 until we reach an n for which no variation n-grams are in the corpus.

Running the variation n-gram algorithm on the WSJ corpus produced variation n-grams up to length 224. The table in Figure 1 reports two results for each n: the first is the number of variation n-grams that were detected and the second is the number of variation nuclei that are contained in those n-grams. For example, the second entry

1.	7033	7033	57.	946	3558	113.	343	1846	169.	90	395
2.	17384	18499	58.	932	3558	114.	338	1820	170.	87	380
3.	12199	13002	59.	918	3557	115.	333	1794	171.	84	365
4.	6576	7181	60.	904	3556	116.	328	1768	172.	81	350
5.	4097	4646	61.	889	3550	117.	323	1742	173.	78	335
6.	2934	3478	62.	873	3545	118.	318	1716	174.	75	320
7.	2333	2870	63.	857	3536	119.	313	1689	175.	72	305
8.	2027	2583	64.	841	3519	120.	308	1661	176.	69	290
9.	1825	2405	65.	825	3497	121.	303	1632	177.	66	274
10.	1678	2296	66.	809	3473	122.	298	1602	178.	63	258
11.	1579	2249	67.	793	3449	123.	293	1571	179.	60	242
12.	1516	2241	68.	777	3426	124.	288	1540	180.	57	226
13.	1475	2260	69.	762	3405	125.	283	1509	181.	54	210
14.	1456	2305	70.	747	3376	126.	278	1478	182.	51	194
15.	1429	2333	71.	733	3348	127.	273	1446	183.	48	178
16.	1413	2378	72.	720	3315	128.	268	1413	184.	45	162
17.	1395	2431	73.	708	3283	129.	263	1379	185.	42	146
18.	1381	2484	74.	696	3250	130.	258	1345	186.	40	137
19.	1376	2547	75.	683	3211	131.	253	1311	187.	38	128
20.	1376	2615	76.	670	3171	132.	248	1277	188.	37	126
21.	1367	2671	77.	656	3134	133.	243	1243	189.	36	124
22.	1355	2721	78.	642	3093	134.	237	1205	190.	35	122
23.	1343	2764	79.	629	3052	135.	231	1167	191.	34	120
24.	1330	2808	80.	616	3011	136.	225	1134	192.	33	118
25.	1318	2846	81.	603	2966	137.	219	1100	193.	32	116
26.	1304	2877	82.	594	2928	138.	213	1066	194.	31	114
27.	1291	2911	83.	585	2890	139.	207	1032	195.	30	112
28.	1283	2950	84.	577	2853	140.	202	1001	196.	29	110
29.	1273	2987	85.	568	2814	141.	197	970	197.	28	108
30.	1264	3028	86.	558	2765	142.	193	948	198.	27	106
31.	1255	3072	87.	547	2714	143.	189	926	199.	26	104
32.	1243	3116	88.	536	2661	144.	185	904	200.	25	102
33.	1234	3164	89.	526	2617	145.	181	882	201.	24	100
34.	1220	3203	90.	517	2573	146.	176	853	202.	23	98
35.	1211	3241	91.	505	2516	147.	171	828	203.	22	96
36.	1201	3275	92.	493	2457	148.	167	809	204.	21	94
37.	1188	3305	93.	481	2398	149.	163	790	205.	20	92
38.	1177	3337	94.	469	2339	150.	159	770	206.	19	90
39.	1169	3371	95.	459	2298	151.	155	750	207.	18	88
40.	1158	3397	96.	449	2259	152.	151	729	208.	17	86
41.	1147	3419	97.	439	2218	153.	147	708	209.	16	84
42.	1134	3432	98.	430	2185	154.	143	687	210.	15	82
43.	1124	3444	99.	421	2150	155.	139	666	211.	14	80
44.	1114	3454	100.	412	2114	156.	135	645	212.	13	78
45.	1106	3468	101.	405	2084	157.	131	623	213.	12	76
46.	1097	3481	102.	399	2066	158.	127	600	214.	11	74
47.	1087	3495	103.	393	2048	159.	123	575	215.	10	72
48.	1074	3503	104.	388	2032	160.	119	550	216.	9	68
49.	1059	3507	105.	383	2017	161.	115	525	217.	8	64
50.	1045	3510	106.	378	2002	162.	111	500	218.	7	59
51.	1030	3510	107.	373	1987	163.	108	485	219.	6	53
52.	1018	3521	108.	368	1969	164.	105	470	220.	5	46
53.	1004	3529	109.	363	1948	165.	102	455	221.	4	38
54.	989	3538	110.	358	1924	166.	99	440	222.	3	29
55.	975	3548	111.	353	1898	167.	96	425	223.	2	20
56.	961	3556	112.	348	1872	168.	93	410	224.	1	10

Figure 1: Variation n-grams and nuclei in the WSJ

reports that 17384 variation bigrams were found, and they contained 18499 variation nuclei, i.e., for some of the bigrams there was a tag variation for both of the words. At the end of the table is the single variation 224-gram, containing 10 different variation nuclei, i.e., spots where the annotation of the (two) occurrences of the 224-gram differ.[5]

[5]The table does not report how often a variation n-gram occurs in a corpus since such a count is not meaningful in our context: The variation unigram *the*, for instance, appears

The table reports the level of variation in the WSJ across identical contexts of different sizes. In the next section we turn to the issue of detecting those occurrences of a variation n-gram for which the variation nucleus is an annotation error.

2.1.2 Heuristics for classifying variation

Once the variation n-grams for a corpus have been computed, heuristics can be employed to classify the variations into errors and ambiguities. The first heuristic encodes the basic fact that the tag assignment for a word is dependent on the context of that word. The second takes into account that natural languages favor the use of local dependencies over non-local ones. Both of these heuristics are independent of a specific corpus, tagset, or language.

Variation nuclei in long n-grams are errors The first heuristic is based on the insight that a variation is more likely to be an error than a true ambiguity if it occurs within a long stretch of otherwise identical material. In other words, the longer the variation n-gram, the more likely that the variation is an error.

For example, *lending* occurs tagged as adjective (JJ) and as common noun (NN) within occurrences of the same 184-gram in the corpus. It is very unlikely that the context (109 identical words to the left, 74 to the right) supports an ambiguity, and the adjective tag does indeed turn out to be an error. Similarly, the already mentioned 224-gram includes 10 different variation nuclei, all of which turn out to be erroneous variation.

While we have based this heuristic solely on the length of the identical context, another factor one could take into account for determining relevant contexts are structural boundaries. A variation nucleus that occurs within a complete, otherwise identical sentence is very likely an error.[6]

For example, the 25-gram in (2) is a complete sentence that appears 14 times, four times with *centennial* tagged as JJ and ten times with *centennial* marked as NN, with the latter being correct according to the tagging guide (Santorini, 1990).

(2) During its *centennial* year , The Wall Street Journal will report events of the past century that stand as milestones of American business history .

Distrust the fringe Turning the spotlight from the n-gram and its properties to the variation nucleus contained in it, an important property determining the likelihood of a variation to be an error is whether the variation nucleus appears at the fringe of the variation n-gram, i.e., at the beginning or the end of the context which is identical over all occurrences.

For example, *joined* occurs as past tense verb (VBD) and as past participle (VBN) within a variation 37-gram. It is the first word in the variation 37-gram and in one of the occurrences it is preceded by *has* and in another it is not. Despite the relatively long context of 37 words to the right, the variation thus is a genuine ambiguity, enabled by the location of the variation nucleus at the left fringe of the variation n-gram.

2.1.3 Results for the WSJ

The variation n-gram algorithm for the WSJ found 2495 distinct variation nuclei of n-grams with $6 \leq n \leq 224$, where by distinct we mean that each corpus position is only taken into account for the longest variation n-gram it occurs in.[7] To evaluate the precision of the variation n-gram algorithm and the heuristics for tag error detection, we need to know which of the variation nuclei detected actually include tag assignments that are real errors. We thus inspected the tags assigned to the 2495 variation nuclei that were detected by the algorithm and marked for each nucleus whether the variation was an error or an ambiguity.[8] We found

56,317 times in the WSJ, but 56,300 of these are correctly annotated as determiner (DT).

[6] Since sentence segmentation information is often available for pos-tagged corpora, we focus on those structural domains here. For treebanks, other constituent structure domains could also be used for the purpose of determining the size of the context of a variation that should be taken into account for distinguishing errors from ambiguities.

[7] This eliminates the effect that each variation n-gram instance also is an instance of a variation (n-1)-gram, a property exemplified by (1) and the discussion below it.

[8] Generally, the context provided by the variation n-gram was sufficient to determine which tag is the correct one for the variation nucleus. In some cases we also considered the wider context of a particular instance of a variation nucleus to verify which tag is correct for that instance. In theory, some of the tagging options for a variation nucleus could be ambiguities, whereas others would be errors; in practice this did not occur.

that 2436 of those variation nuclei are errors, i.e., the variation in the tagging of those words as part of the particular n-gram was incorrect. To get an idea for how many tokens in the corpus correspond to the 2436 variation nuclei that our method correctly flagged as being wrongly tagged, we hand-corrected the mistagged instances of those words. This resulted in a total of 4417 tag corrections.

Turning to the heuristics discussed in the previous section, for the first one an n-gram length of six turns out to be a good cut-off point for the WSJ. This becomes apparent when one takes a look at where the 59 ambiguous variation nuclei arise: 32 of them are variation nuclei of 6-grams, 10 are part of 7-grams, 4 are part of 8-grams, and the remaining 13 occur in longer n-grams.

Regarding the second heuristic, distrust the fringe, 57 of the 59 ambiguous variation nuclei that were found are fringe elements, i.e., occur as the first or last element of the variation n-gram. The two exceptions are "*and **use** some of the proceeds to*" and "*buy and **sell** big blocks of*", where the variation nuclei *use* and *sell* are ambiguous between base form verb (VB) and third-person singular present tense verb (VBP) but do not occur at the fringe. As an interesting aside, more than half of the true ambiguities (31 of 59) occurred between past tense verb (VBD) and past participle (VBN) and are the first word in their n-gram.

Problematic cases Of the 2436 erroneous variation nuclei we discussed above, 140 of them deserve special attention here in that it was clear that the variation was incorrect, but it was not possible to decide based on the tagging guide (Santorini, 1990) which tag would be the right one to assign.[9] That is, even without knowing the correct tag, it is clear that the context demands a uniform tag assignment. Most of those cases concern the distinction between singular proper noun (NNP) and plural proper noun (NNPS). For example, in the bigram *Salomon Brothers*, *Brothers* is tagged 42 times as NNP and 30 times as NNPS; similarly, *Motors* in *General Motors* is an NNP 35 times and an NNPS 51 times.

While these variation nuclei clearly involve erroneous variation, they were not included in the total count of incorrect tag assignments detected by the variation n-gram method since the number to be added depends on which tag is deemed to be the correct one. For the NNP/NNPS cases, either there are 362 additional errors in the corpus (if NNP is correct) or 369 additional ones (in the other case).

2.2 Two simple ideas

Aside from the main proposal of this paper, to use a variation n-gram analysis combined with heuristics for detecting corpus errors, there are two simple ideas for detecting errors which we want to mention here. These techniques are conceptually independent of the variation n-gram method, but can be combined with it in a pipeline model.

2.2.1 Closed class analysis

Lexical categories in linguistics are traditionally divided into open and closed classes. Closed classes are the ones for which the elements can be enumerated (e.g., classes like determiners, prepositions, modal verbs, or auxiliaries), whereas open classes are the large, productive categories such as verbs, nouns, or adjectives.

Making practical use of the concept of a closed class, one can see that almost half of the tags in the WSJ tagset correspond to closed lexical classes. This means that a straightforward way for checking the assignment of those tags is available. One can search for all occurrences of a closed class tag and verify whether each word found in this way is actually a member of that closed class. This can be done fully automatically, based on a list of tags corresponding to closed classes and a list of the few elements contained in each closed class.[10]

The WSJ annotation uses 48 tags (incl. punctuation tags), of which 27 are closed class items. Searching for determiners (DT) we found 50 words that were incorrectly assigned this tag. Ex-

[9] While this is a problem with the pos-annotation in the Penn Treebank, Voutilainen and Järvinen (1995) show that in principle it is possible to design and document a tagset in a way that allows for 100% interjudge agreement for morphological (incl. part-of-speech) annotation.

[10] Conversely, one can also search for all occurrences of a particular word that is a member of a closed class and check that only the closed class tag is assigned. Some of these words are actually ambiguous, though, so that additional lexical information would be needed to correctly allow for additional tag assignments for such ambiguous words.

amples for the mistagged items include *half* in both adjectival (JJ) and noun (NN) uses, the predeterminer (PDT) *nary*, and the pronoun (PRP) *them*. Looking through three closed classes, we detected 94 such tagging errors.

In sum, such a closed class analysis seems to be useful as an error detection/correction method, which can be fully automated and requires very little in terms of language specific resources.

2.2.2 Implementing tagging guide rules

Baker (1997) discusses that the BNC Tag Enhancement Project used context sensitive rules to fix annotation errors. The rules were written by hand, based on an inspection of errors that often resulted from the focus of the automatic tagger on few properties in a small window. Oliva (2001) also discusses building and applying such rules to detect potential errors; some rules are specified to automatically correct an error, while others require human intervention.

Tagging guides such as the one for the WSJ (Santorini, 1990) often specify a number of specific patterns and state explicitly how they should be treated. One can therefore use the same technology as Baker (1997), Oliva (2001) and others and write rules which match the specific patterns given in the manual, check whether the correct tags were assigned, and correct them where necessary. This provides valuable feedback as to how well the rules of the tagging guide were followed by the corpus annotators and allows for the automatic identification and correction of a large number of error pattern occurrences.

For example, the WSJ tagging manual states: "Hyphenated nominal modifiers ... should always be tagged as adjectives." (Santorini, 1990, p. 12). While this rule is obeyed for 8605 occurrences in the WSJ, there are also 2466 cases of hyphenated words tagged as nouns preceding nouns, most of which are violations of the above tagging manual guideline, such as, for instance, *stock-index* in *stock-index futures*, which is tagged 41 times as JJ and 36 times as NN.

3 Related work

Considering the significant effort that has been put into obtaining pos-tagged reference corpora in the past decade, there are surprisingly few publications on the issue of detecting errors in pos-annotation. In the past two or three years, though, some work on the topic has appeared, so in the following we embed our work in this context.

The starting point of our variation n-gram approach, that variation in annotation can indicate an annotation error, essentially is also the starting point of the approach to annotation error detection of van Halteren (2000). But while we look for variation in the annotation of comparable stretches of material within the corpus, Van Halteren proposes to compare the hand-corrected annotation of the corpus with that produced by an automatic tagger, based on the idea that automatic taggers are designed to detect "consistent behavior in order to replicate it".[11] Places where the automatic tagger and the original annotation disagree are thus deemed likely to be inconsistencies in the original annotation. Van Halteren shows that his idea is successful in locating a number of potential problem areas, but he concludes that checking 6326 areas of disagreement only unearths 1296 errors. The precision for detecting errors based on tagger-annotation disagreement thus is rather low, which is problematic considering that the repair stage that weeds out the many false positives of error detection is a manual process.

Eskin (2000) discusses how to use a sparse Markov transducer as a method for what he calls anomaly detection. The notion of an anomaly essentially refers to a rare local tag pattern. The method flags 7055 anomalies for the Penn Treebank, about 44% of which hand inspection shows to be errors. Just as discussed for the approach of Van Halteren mentioned above, the low precision of the method of Eskin for detecting errors means that the repair process has to deal with a high number of false positives from the detection stage, which is problematic since error correction is done manually. In terms of the kind of errors that are detected by the sparse Markov transducer, Eskin notes that "if there are inconsistencies between annotators, the method would not detect the errors

[11] Abney et al. (1999) suggest a related idea based on using the importance weights that a boosting algorithm employed for tagging assigns to training examples; but they do not explore and evaluate such a method.

because the errors would be manifested over a significant portion of the corpus." Eskin's method thus nicely complements the approach presented in this paper, given that inter-annotator (and intra-annotator) errors are precisely the kinds of errors our variation n-gram method is designed to detect.

Květǒn and Oliva (2002) employ the notion of an invalid bigram to locate corpus positions with annotation errors. An invalid bigram is a pos-tag sequence that cannot occur in a corpus, and the set of invalid bigrams is derived from the set of possible bigrams occurring in a hand-cleaned sub-corpus, as well as linguistic intuition. Using this method, Květǒn and Oliva (2002) report finding 2661 errors in the NEGRA corpus (containing 396,309 tokens). Interestingly, most of the errors found by the approaches we presented in this paper are perfectly valid bigrams. The invalid bigram approach of Květǒn and Oliva (2002) thus also nicely complements our proposal.

Hirakawa et al. (2000) and Müller and Ule (2002) are two approaches which use the pos-annotation as input for syntactic processing—a full syntactic analysis in the former and a shallow topological field parse in the latter case—and single out those sentences for which the syntactic processing does not provide the expected result. Different from the approach we have described in this paper, both of these approaches require a sophisticated, language specific grammar and a robust syntactic processing regime so that the failure of an analysis can confidently be attributed to an error in the input and not an error in the grammar or the processor.

4 Summary and Outlook

We have presented three detection methods for pos-annotation errors which remain in gold-standard corpora despite human post-editing. Our main proposal is to detect variation within comparable contexts and classify such variation as error or ambiguity using heuristics based on the nature of the context. The detection method can be automated, is independent of the particular language and tagset of the corpus, and requires no additional language resources such as lexica. We showed that an instance of this method based on identity of words in the variation contexts, so-called variation n-grams, successfully detects a variety of errors in the WSJ corpus.

The usefulness of the notion of a variation n-gram relies on a particular word to appear several times in a corpus, with different annotations. It thus works best for large corpora and hand-annotated or hand-corrected corpora, or corpora involving other sources of inconsistency. As Ratnaparkhi (1996) points out, interannotator bias creates inconsistencies which a completely automatically-tagged corpus does not have. And Baker (1997) makes the point that a human post-editor also decreases the internal consistency of the tagged data since he will spot a mistake made by an automatic tagger for some but not all of its occurrences. As a result, our variation n-gram approach is well suited for the gold-standard annotations generally resulting from a combination of automatic annotation and manual post-editing. A case in point is that we recently applied the variation n-gram algorithm to the BNC-sampler corpus and obtained a significant number of variation n-grams up to length 692.

The variation n-gram approach as the instance of our general idea to detect variation in comparable contexts presented in this paper prioritizes the precision of error detection by requiring identity of the words in the context of a variation in order for a variation n-gram to be detected. Despite this emphasis on precision, the significant number of errors the method detected in the WSJ shows that the recall obtained is useful in practice. In the future, we intend to experiment with defining variation contexts based on other, more general properties than the words themselves in order to increase recall, i.e., the number of errors detected. Natural candidates are the pos-tags of the words in the context. Other context generalizations also seem to be available if one is willing to include language or corpus specific information in computing the contexts. In the WSJ corpus, for example, different numerical amounts, which frequently appear in the same context, could be treated identically.

In terms of outlook, the variation n-gram method can also be applied to other types of corpus annotation. Given that the quality of syntactic constituency and function annotation in current treebanks lags significantly behind that of pos-

annotation, methods for detecting errors in syntactic annotation have a wide area of application. By applying the variation n-gram method to a syntactically-annotated string, we can detect those n-grams which occur several times but with a different constituent structure or syntactic function. Future research has to show whether it is possible to classify the syntactic variation n-grams thus detected into errors and ambiguities with the same precision as is the case for the pos-annotation variation n-grams we discussed in this paper.

Acknowledgements We would like to thank the anonymous reviewers of EACL and LINC for their comments and the participants of the OSU computational linguistics discussion group CLippers.

References

Steven Abney, Robert E. Schapire and Yoram Singer, 1999. Boosting Applied to Tagging and PP Attachment. In Pascale Fung and Joe Zhou (eds.), *Proceedings of Joint EMNLP and Very Large Corpora Conference*. pp. 38–45.

Rakesh Agrawal and Ramakrishnan Srikant, 1994. Fast Algorithms for Mining Association Rules in Large Databases. In Jorge B. Bocca, Matthias Jarke and Carlo Zaniolo (eds.), *VLDB'94*. Morgan Kaufmann, pp. 487–499.

John Paul Baker, 1997. Consistency and accuracy in correcting automatically tagged data. In Garside et al. (1997), pp. 243–250.

Thorsten Brants, 2000. Inter-Annotator Agreement for a German Newspaper Corpus. In *Proceedings of LREC*. Athens, Greece.

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

Roger Garside, Geoffrey Leech and Tony McEnery (eds.), 1997. *Corpus annotation: linguistic information from computer text corpora*. Longman, London and New York.

Hideki Hirakawa, Kenji Ono and Yumiko Yoshimura, 2000. Automatic Refinement of a POS Tagger Using a Reliable Parser and Plain Text Corpora. In *Proceedings of COLING*. Saarbrücken, Germany.

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

Geoffrey Leech, 1997. *A Brief Users' Guide to the Grammatical Tagging of the British National Corpus*. UCREL, Lancaster University.

Geoffrey Leech, Roger Garside and Michael Bryant, 1994. CLAWS4: The tagging of the British National Corpus. In *Proceedings of COLING*. Kyoto, Japan, pp. 622–628.

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.

Frank H. Müller and Tylman Ule, 2002. Annotating topological fields and chunks – and revising POS tags at the same time. In *Proceedings of COLING*. Taipei, Taiwan.

Karel Oliva, 2001. The Possibilities of Automatic Detection/Correction of Errors in Tagged Corpora: A Pilot Study on a German Corpus. In Václav Matoušek, Pavel Mautner, Roman Mouček and Karel Taušer (eds.), *Text, Speech and Dialogue (TSD)*. Springer, pp. 39–46.

Adwait Ratnaparkhi, 1996. A maximum entropy model part-of-speech tagger. In *Proceedings of EMNLP*. Philadelphia, PA, pp. 133–141.

Beatrice Santorini, 1990. Part-Of-Speech Tagging Guidelines for the Penn Treebank Project (3rd revision, 2nd printing). Ms., Department of Linguistics, UPenn. Philadelphia, PA.

John M. Sinclair, 1992. The automatic analysis of corpora. In Jan Svartvik (ed.), *Directions in Corpus Linguistics*, Mouton de Gruyter, Berlin and New York, NY, pp. 379–397.

Wojciech Skut, Brigitte Krenn, Thorsten Brants and Hans Uszkoreit, 1997. An Annotation Scheme for Free Word Order Languages. In *Proceedings of ANLP*. 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 the 2nd Workshop on Linguistically Interpreted Corpora*. Luxembourg.

Atro Voutilainen and Timo Järvinen, 1995. Specifying a shallow grammatical representation for parsing purposes. In *Proceedings of the 7th Conference of the EACL*. Dublin, Ireland.