Similarity and Dissimilarity in Treebank Grammars

Markus Dickinson
Indiana University
317 Memorial Hall
Bloomington, IN 47405
md7@indiana.edu

Abstract

To uncover rules in a treebank grammar which are of dubious quality, we investigate two methods for detecting problematic structures, both based on the same notion of similarity. The first is based on the notion that similar rules should receive the same annotation. The second is based on the idea that rules which are dissimilar to other rules are likely problematic. We show these two methods to be effective in detecting erroneous rules, rules used for ungrammatical or otherwise non-standard constructions, and rules which reveal non-uniform decisions made in the annotation scheme.

1 Introduction and Motivation

While annotated corpora are commonly used for both natural language processing (NLP) and for linguistic searching, there is a need to investigate the quality of such annotation. Corpora can be viewed as large repositories of language data, useful for the construction and validation of linguistic theories. As such, there is an increasing number of linguistically-annotated corpora, presenting information on a wide range of linguistic properties, such as morphological distinctions (e.g., Leech, 1997), syntactic distinctions (e.g., Sampson, 1995), semantic distinctions (e.g., Kingsbury et al., 2002), and discourse distinctions (e.g., Allen and Core, 1996). This annotation present in a corpus is the result of applying an annotation scheme to the data, capturing the desired distinctions. In other words, annotation encodes a linguistic description of the data, and in order for this to be useful, it must be of high quality.

When we speak of a theory being encoded in a treebank, of course, it is not that simple. There are competing factors in what kind of (descriptive) linguistic theory is encoded. On the one hand, corpus annotation is guided by external criteria: do the distinctions capture linguistic properties needed for certain corpus uses, such as parsing or linguistic searching? On the other hand, we have internal criteria: can the distinctions be annotated easily and automatically (cf. Elworthy, 1995; Déjean, 2000)?

Syntactically-annotated corpora, or treebanks, tend to emphasize broad coverage and these so-called internal criteria, making sure that the annotation can be done consistently, with high inter-annotator agreement (e.g., Voutilainen and Järvinen, 1995). This is often at the expense of true grammar development, however. While the treebank is annotated quickly and in a way which lends itself to parsing, it is not clear what the properties are of the encoded grammar. Does it match anything resembling linguistic theory, or was it annotated consistently at the cost of being theoretically desirable?
As one example, consider the partial tree from the Wall Street Journal (WSJ) corpus portion of the Penn Treebank (PTB, Marcus et al., 1993) shown in 1. To avoid disagreements between annotators, the trees are given flat structures; in this case, no one has to decide where the *as* phrase attaches, instead including it simply as a daughter of the VP. Thus, we wind up with rules like \( VP \rightarrow VB \ NP \ PP \ NP \), which do not correspond to theories distinguishing arguments from adjuncts in the syntactic structure.

\[
\begin{array}{c}
\text{VP} \\
\text{VB} \quad \text{NP}
\end{array}
\]

\[
\begin{array}{c}
\text{join} \\
\text{DT} \quad \text{NN} \\
\text{the} \quad \text{board}
\end{array}
\]

\[
\begin{array}{c}
\text{PP} \\
\text{IN} \quad \text{NP}
\end{array}
\]

\[
\begin{array}{c}
\text{as} \\
\text{DT} \quad \text{JJ} \quad \text{NN}
\end{array}
\]

\[
\begin{array}{c}
\text{NP} \\
\text{NNP} \quad \text{CD} \\
\text{Nov.} \quad 29
\end{array}
\]

\[
\begin{array}{c}
\text{NP} \\
\text{DT} \quad \text{JJ} \quad \text{NN}
\end{array}
\]

\[
\begin{array}{c}
\text{a} \quad \text{nonexecutive} \quad \text{director}
\end{array}
\]

Although such treebanks have served to advance the state-of-the-art in computational linguistics, there are still problems with using grammars extracted from them, regardless of whether we intend to extract a grammar for parsing or for linguistic analysis. First, as illustrated above, treebanks commonly contain rather flat structures and coarse categories. This means that there are missing linguistic decisions, which would have to be recovered for linguistic searching. Indeed, these distinctions are often useful to use in parser models and must be recovered for parser training (cf. Petrov et al., 2006). Furthermore, the distinctions may not only be missing, but may be incompatible with a linguistic theory. Secondly, there is the sheer number of rules to contend with. The WSJ, for example, has over 17,000 rules for 50,000 sentences. Grammar compaction methods can reduce the size of the rule set (Krotov et al., 1998; Hepple and van Genabith, 2000), but there is still a need to sort useful rare constructions from unhelpful ones (Foth and Menzel, 2006; Daelemans et al., 1999). Finally, there is the problem of annotation errors, which arise in the process of creating a large corpus. These errors have a detrimental effect on the training and evaluation of natural language processing (NLP) systems (cf. Dickinson and Meurers, 2005a; Hogan, 2007) and also on the precision and recall for finding desired linguistic constructions (cf. Meurers, 2005). For example, Padro and Marquez (1998) show that, for many current comparative evaluation situations, one cannot truly tell which technology is better.

These three problems are related, in that they deal with the quality of corpus annotation. To attack these problems, we set out to investigate and automatically identify problematic treebank rules, which reveal different quality issues with the treebank. Namely, problematic rules might be erroneous, cover ungrammatical constructions, or reveal the quirks of an annotation scheme, that is, where it needs feedback.

But how can one investigate rule quality in an automatic way? How can one rule be better or worse than another? To answer this, we use properties of the whole grammar itself as a guide to how rules should generally be organized. Specifically, we focus on finding similarities and dissimilarities among rules, and we make two hypotheses, which
we show to be effective. The first hypothesis is that similar rules should receive the same annotation, as discussed in section 3. The more similar two rules are, the more we expect them to be categorized in the same way. The second hypothesis is that rules which are dissimilar to every other rule are likely problematic, as outlined in section 4. If a rule behaves like nothing else, there should be a good reason for it; if there is no reason, there is likely a problem. For both approaches, the same definition of similarity can be used.

While we only investigate problematic rules, finding commonalities between rules provides insights into an appropriate syntactic model for treebank grammars. This is an important step for tasks such as recovering latent annotation, grammar compaction, and annotation scheme revision.

2 Background

To motivate the need for a definition of similarity across treebank rules, we start with an error detection method that searches for inconsistency of labeling within local trees (Dickinson and Meurers, 2005b). The insight is that one can generally determine the syntactic category of the mother of a rule based on the categories of its daughters. In other words, linguistic phrase structure rules tend to be endocentric (cf. X-bar syntax, Jackendoff, 1977). If the same daughters list has more than one mother, this might indicate a violation of endocentricity. Thus, Dickinson and Meurers (2005b) search for variation in mother categories which dominate the same daughters; daughters lists with more than one mother are flagged as potential errors. This method turns out to be quite successful at detecting errors, as 74% of variations in the WSJ contain errors.

As an example, consider the daughters list JJ , NN CC JJ, which varies between unlike coordinated phrase (UCP) and adjective phrase (ADJP), as in (2). Here, we successfully flag an error, as there is no need for variation: the guidelines indicate that ADJP is erroneous Bies et al. (1995, p. 120).

\( (2) \begin{align*}
 & \text{a. [}_{UCP}\text{ federal/JJ },/\text{, state/NN and/CC local/JJ]} \text{ public officials} \\
 & \quad \text{UCP} \\
 & \quad \quad \text{JJ , NN CC JJ} \\
 & \quad \quad \quad \text{federal , state and local} \\
 & \quad \text{b. [}_{ADJP}\text{ scientific/JJ },/\text{, engineering/NN and/CC academic/JJ]} \text{ communities} \\
 & \quad \text{ADJP} \\
 & \quad \quad \text{JJ , NN CC JJ} \\
 & \quad \quad \quad \text{scientific , engineering and academic}
\end{align*} \)

A limitation of the method is its lack of generality, in that a rule occurring once cannot vary with any other rule (Dickinson, 2006). Furthermore, it is not only identical daughters lists which must share the same mother, but also very similar daughters lists. In order to increase recall, we note that many rare rules are actually quite similar to other rules. Currently, for example, the daughters lists ADVP RB ADVP and ADVP , RB ADVP, as shown in (3), are treated distinctly. If they are treated as the same daughters list, then
there are two different mothers, PP (prepositional phrase) and ADVP (adverbial phrase), and this variation points to the presence of an error, in PP in this case. What is needed is a way to say that these rules are behaving in the same way.

(3) a. to slash its work force in the U.S., [PP [ADVP as] soon/RB [ADVP as next month]]
   b. to report their purchases and sales [ADVP [ADVP immediately] J, not/RB [ADVP a month later]]

3 Similarity

To increase the number of errors found, therefore, we relax the identity requirement between daughters lists. Namely, we automatically create equivalence classes of daughters lists to assist in error detection, where the defining property of making equivalences is that the mother of each rule is still predicted to be the same by all daughters list in the class. To create such equivalences, we can use the insight that anything not contributing to prediction can be ignored. Grouping rules based on the predictive information of daughters will result in better recall of errors.

3.1 Equivalence criteria

Based on this notion of predictability, we can use fairly simple properties of rules in order to establish equivalences between them. For each daughters list, the steps we employ are the following (Dickinson, 2006):

1. Remove daughter categories that are always non-predictive to phrase categorization.
2. Group head-equivalent lexical categories.
3. Model adjacent identical elements as a single element.

In the first step, we remove daughter categories that are always non-predictive to phrase categorization, and these are categories which are always adjuncts. Clearly, we need to keep the head in a list of daughters, as head information generally percolates up to the mother. We also need to keep all arguments, as arguments can distinguish one rule from another. For example, SINV (inverted sentence) is the mother for the daughters list VP NP PP because the head (VP, verb phrase) precedes the argument (NP, noun phrase). We thus want to remove adjuncts, but this has to be done in such a way that it generalizes across all rules. Thus, we focus on eliminating “inherent” adjuncts, which come in a few different forms in the PTB. First, there is punctuation, which is not involved in the predicate-argument structure (cf. Hollingshead et al., 2005), and is often not included in treebank rules (cf. Brants et al., 2002). Secondly, parentheticals (PRN) refer to solely parenthetical material and thus are always adjuncts when they occur in a daughters list. Finally, empty elements (i.e., -NONE-) can refer to any category, and thus they are uninformative with respect to the mother category.¹

¹Note that empty elements are generally a problem for error detection (Dickinson and Meurers, 2003).
The second step is to group what we call *head-equivalent* lexical categories, or categories which are the same in predicting the mother. The intuition here is that phrases headed by, for example, either singular common noun (NN) or plural common noun (NNS) are generally NP, and this distinction does not add any information in predicting the mother. The full set of mappings is given in table 1, which is similar to mapping #2 in Hepple and van Genabith (2000). All other lexical tags (i.e., not in the table) are not grouped with any other tags: there are 13 such tags only equivalent to themselves.

<table>
<thead>
<tr>
<th>Base category</th>
<th>Head Equivalence Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determiners</td>
<td>{DT, PDT, PRP}</td>
</tr>
<tr>
<td>Adjectives</td>
<td>{JJ, JJR, JJS}</td>
</tr>
<tr>
<td>Nouns</td>
<td>{NN, NNS, PRP}</td>
</tr>
<tr>
<td>Proper nouns</td>
<td>{NNP, NNPS}</td>
</tr>
<tr>
<td>Adverbs</td>
<td>{RB, RBR, RBS}</td>
</tr>
<tr>
<td>Verbs</td>
<td>{MD, VB, VBD, VBG, VBN, VBP, VBZ}</td>
</tr>
<tr>
<td>Wh-determiners</td>
<td>{WDT, WP}</td>
</tr>
</tbody>
</table>

Table 1: Head-equivalence classes in the WSJ

The final step is to model adjacent identical elements as a single element. This is akin to modeling a flat series of identical categories with the Kleene + operator (XP+), and we do this for arbitrarily long sequences. For instance, for the daughters list NN IN NP IN NP, the second IN NP says nothing more about predicting mother category, and so this is mapped to NN IN NP. All repetitious sequences are mapped into the shortest possible sequence that is still predictive. For JJ NN NN JJ NN, for example, we first reduce it to JJ NN JJ NN, and then to JJ NN. This Kleene reduction step bears much in spirit to the correcting of illegal syntactic structures for the Penn Korean Treebank in Han et al. (2002), where legitimate right-hand sides of rules are hand-encoded as regular expressions, in order to detect and correct illegal syntactic structures.

### 3.2 Results

**Evaluation of equivalence criteria** Before testing the effect of grouping rules into equivalence classes for error detection, we first want to ensure that the equivalence criteria are not eliminating anything essential to phrase categorization. In other words, do we lose any predictive information by using equivalence classes? To gauge this, we sampled 100 equivalence classes from section 00 of the WSJ and examined them by hand. It turns out that in 98 cases, nothing predictive has been removed. In fact, in 24 of those cases, the mapping is complete; that is, nothing more should have been removed.

The two unsuccessful mappings involve the label NAC (not a constituent), as illustrated in (4). The daughters list NNP , NNP , reduces to NNP, but NNP by itself does not predict NAC. In fact, Bies et al. (1995, p. 208) mention that the presence of a comma partially determines NAC here. In the future, one could consider being more careful with punctuation, but the results are effective enough for us to continue.

(4) $[_{\text{NAC}} \text{Albuquerque/NNP}, \text{N.M./NNP} ,]$
Results for rule similarity  Turning to the results for the rule similarity error detection method, by grouping daugthers lists into equivalence class, we map 15,989 daughters lists to 3783 equivalence classes. From these 3783 classes, 546 have variation in their labeling. While 546 is less than the original 844 varying daughters lists, each class is much bigger. In fact, since all instances of an original rule are within the same class, we are guaranteed to find at least as many varying rules as before.

We sampled 100 of these 546 variations and marked for each whether it contained an error, and we thus estimate 71% error detection precision. Significantly, this is on a par with the original error detection precision (74%), which means that we are increasing recall without sacrificing precision.

Indeed, this method points to nearly a thousand erroneous rule types, and these are a superset of the cases in Dickinson and Meurers (2005b). As one example, consider the equivalence class JJ VB, as shown in examples (5) and (6). Here, we have four different daughter lists all mapping to JJ VB, with variation between ADJP and NP. Crucially, without equivalence classes, the list JJR VBN, with ADJP as its mother category, would not have been found to be in variation with NP when using daughters lists.

(5) a. \[NP\] personal-income/JJ growth/VB]
   b. \[NP\] other/JJ high-yield/JJ deals/VBZ]
   c. \[NP\] last/JJ May/MD]
(6) \[ADJP\] lesser/JJR developed/VBN]

Thus, we successfully increase error detection recall, without sacrificing precision, and we do so with only a small amount of manual work.

While this is successful for the WSJ, new criteria would have to be hand-written for other treebanks and languages. The same general principles apply, namely maintaining the predictability of the mother category. The only caveat is that one must be careful about which abstractions to make; for example, case distinctions important for subcategorization should not be conflated.

Annotation scheme issues  Detecting errors is not all that these method allows us to do. This method of using rule similarity also reveals various issues with the annotation scheme. We provide a few brief examples here, as a way to illustrate the types of cases that such methods uncover.

First, as before, we are able to turn up cases of acceptable non-endocentricity, as in example (7), where a PP is headed by a verb. We also find more subtle cases where the annotators had difficulty in deciding what type of phrase a sequence should be categorized as. For example, we find nominal constructions functioning adverbially, as in (8), with ADVPs apparently headed by nominals, apparent violations of endocentricity whose annotation is not entirely clear. Finally, we find inconsistent treatment of undocumented constructions, for instance, *due to* phrases.

(7) \[PP\] not/RB including/VBG \[NP\] the Soviet Union]]

(8) a. \[ADVP\] way/NN sky/RB] high
   b. \[ADVP\] \[NP\] a week] ago/RB \[NP\] Friday]]
4 Dissimilarity

Using equivalence classes allows us to compare the similarity of rules, but looking at the comparability of treebank rules has a broader effect. So far, equivalence classes provide a way to talk about rule similarity; the other side is to talk about rule dissimilarity. Equivalence classes can be used to provide support for the validity of a rule: the more rules that are within a class, the more evidence that the annotation scheme legitimately licenses that sequence.

Or, to view this from the opposite direction, a rule which does not have any equivalents in the grammar is more likely to be covering a non-linguistic construction. This is indeed worth investigating, as 2141 of the 3783 equivalence classes in the WSJ have only one unique daughters list. Consider an example like (9): here, the daughters list RB TO JJ NNS has no similar rules in the treebank, and it indeed is erroneous (Bies et al., 1995, p. 179).

(9) [NP close/RB to/TO wholesale/JJ prices/NNS]

From a random sample of 100 of the 2141 unique rules, we find that 39 of them are errors. Thus, we estimate 835 errors, with a 95% confidence interval of 630 to 1040 errors. Additionally, we find four rules which cover ungrammatical sentences.²

This 39% precision is rather low, but recall that we essentially get these dissimilar cases for free, in that the same notion of similarity is used here as in the endocentricity-based method. Furthermore, as we will see below, there are interesting reasons other than errors for rules being anomalous.

Additionally, we can identify a subset of cases which is more likely to contain errors. The notion of dissimilarity is based on the notion that daughters lists can and should vary (e.g., by having more or less adjuncts), but this is not always the case. Some rules correctly allow for no fluctuation; for example, the rule NP → EX for the existential use of there (with 1075 occurrences) cannot be modified in any way. It turns out that 777 of the 2141 unique rules map to themselves; that is, nothing was changed in creating the equivalence class. But the other 1364 rules did change, more strongly indicating that they could potentially fluctuate: they map to something that could have been comparable but does not exist in the treebank. Sampling 100 of these 1364 cases turns up 49 errors and 6 rules for ungrammatical sentences. In the future, one could experiment with more carefully identifying daughters lists which are rightfully incomparable to other rules.

We can finally note that the errors we detect overlap very little with the errors detected in the similarity method. Of the 1364 cases, only five have more than one mother. Furthermore, because we are looking at unlikely sequences of daughters, we tend to find more bracketing errors than in the first method. So, even though the methods share the same notion of equivalence, they lead to finding different types of errors because they look for different types of anomalies.

²While the term “ungrammatical” can be debatable, we use it in instances where the sentence is simply non-English; see, e.g., Foster (2007) for more discussion of grammaticality and the need to distinguish it in natural language parsing.
4.1 Analyzing the cases

To understand what this method does, it is instructive to walk through the different kinds of cases that the method turns up, both for dissimilarities which highlight errors or cover ungrammatical sentences and for those which are correct, as they reveal interesting properties of the annotation scheme. We thus outline the three kinds of cases here.

Errors We have established that there are errors in the treebank, but a major question is, why are there errors? As it turns out in this case, one of the major sources of errors is the overapplication of flat structures in the treebank. In the WSJ, many constructions are supposed to be left flat, such as nominal modifiers which are themselves nouns, but this does not apply to all cases. In (10), for example, the RB JJ sequence annotated as in (10a) should be bracketed as an ADJP, as in (10b). We find these cases because we have the sequence is more anomalous without that extra layer of structure.

\[
\text{(10) there seems } * \text{ to be } [\text{NP a/DT fairly/RB systematic/JJ effort/NN } [S * \text{ to address the problem}]]
\]

\[
\text{a.}
\]

\[
\begin{align*}
\text{NP} & \quad \text{DT} \\
& \quad \text{RB} \\
& \quad \text{JJ} \\
& \quad \text{NN} \\
& \quad S
\end{align*}
\]

\[
\text{to address the problem}
\]

\[
\text{b.}
\]

\[
\begin{align*}
\text{NP} & \quad \text{DT} \\
& \quad \text{ADJP} \\
& \quad \text{NN} \\
& \quad S
\end{align*}
\]

\[
\text{to address the problem}
\]

Likewise, the dissimilarity method finds categories which appear in the wrong context. In (11), for instance, the category DT (determiner) appears in the daughters list DT ADJP CC PP, which should be CC ADJP CC PP. Finding a DT next to an ADJP is normally not a problem, but given the context of the whole rule (where CC [coordinating conjunction] is present), we are able to detect a misplaced category.

\[
\text{(11) } [\text{UCP both/DT } [\text{ADJP prudent}] \text{ and/CC } [\text{PP in the best long-term interest of the shareholders}]]
\]

Finally, we even detect some non-endocentric structures with the dissimilarity method, as in (12), where the POS (possessive ending) category appears to be heading a VP. The reason we detect these cases is that these are also categories in the wrong context. POS NP SBAR is not a normal sequence, irrespective of its mother category, which in this case does not match.
(12) It \( [VP \, ’s/POS [NP \, that \, last \, set \, of \, numbers \, , \, that \, *T* \, gives \, the \, Giants \, hope \, in \, the \, Series \, games \, 0 \, *T* \, to \, come] \) .

**Ungrammatical constructions** In addition to errors, this method unconverts constructions which are for ungrammatical language, or non-standard English. For example, in (13), we see rules such as \( QP \rightarrow RB \, JJ \, $ \, CD \), which are the best analyses given the annotation scheme, but which are clearly for ill-formed constructions (e.g., *as little $ 3 should be as little as $ 3*). Likewise, as shown in (14), there are rules which are used to cover “financialspeak.” In either case, it is not clear that these types of rules should be annotated in the same fashion as grammatical rules.

(13) a. Now, they ’re charging \( [QP \, as/RB \, little/JJ \, $$/3/CD] \, *U* \, a \, day \).
   
   b. Chemical earnings \( [VP \, declined/VBD \, [PP \, by \, one-third]\, [PP \, to \, $ \, 120 \, million \, *U*]\, [X \, last \, year \, ’s \, robust \, levels]\) ]

(14) a. \( [NP \, [NP \, Net \, income\, [x \, *]:/: \, [NP \, $ \, 599.9 \, million \, *U* \, ; \, or \, $ \, 20.20 \, *U* \, a \, share]\, ] \)
   
   b. \( [FRAG \, [NP \, Call\, [PP \, at \, par]\, [PP \, after \, two \, years]\, and/CC \, [ADV \, Permission \, thereafter]\, [PP \, at \, par]\, [NP \, every \, six \, months\, ]/.\] \)

**Annotation scheme & guidelines** Finally, we turn to some correct corpus examples, where the method uncovers properties of the annotation scheme and the guidelines which seems to be reflective of non-uniform practices. For treebanking purposes, this is perhaps the most useful aspect of the dissimilarity method: it reveals properties of the annotation which need to be revisited. We present only a few examples here, although there are more which are discovered.

First, we find issues with the category \( QP \) (quantifier phrase), used for complex numerical determiners. According to the guidelines, both *as little as* and ranges of dollar amounts are annotated as flat QPs (Bies et al., 1995, p. 194-202). Crucially, though, there is no guidance on what to do when they are together, and so we find flat QP structures, as in (15).

(15) \( [QP \, as/RB \, little/JJ \, $$/89/CD \, to/TO \, $$/109/CD] \)

Secondly, we have cases like (16) which arguably contains a parenthetical, namely the string *and however unfairly*. Since parentheticals (PRN) are “determined ultimately by individual annotator intuition” (Bies et al., 1995, p. 50), we cannot say that this analysis is incorrect. We can note, however, that with a parenthetical analysis, the daughters list PP PRN NP VP would reduce to the frequent PP NP VP.

(16) \( [S \, [PP \, Like \, Lebanon]\, \, , \, and/CC \, [ADV \, however \, unfairly]\, \, , \, [NP \, Israel]\, \, [VP \, is \, regarded \, * \, by \, the \, Arab \, world \, as \, a \, colonial \, aberration]\, ]/.\] \)
Relatedly, sometimes one analysis is not the absolutely correct one, but only preferred over another analysis. For example, VP gapping is the preferred analysis in some structures, but variability is explicitly allowed for (Bies et al., 1995, p. 125). Thus, we find odd rules such as \( NP \rightarrow NP \), \( ADVP RB NN \), which could also have been analyzed as having a gapped verb.

(17) they do not serve \([NP \ [NP \ the \ people] \ /, \ and/CC \ [ADVP \ particularly] \ not/RB \ consumers/NN] \).  

5 Possible extensions

The methods we have described are clearly effective for detecting errors and other anomalous rules in a flat treebank, but they can be generalized for work on this and other treebanks.

5.1 Generalizing the search for anomalous rules

We have treated the dissimilarity method essentially as a side effect of the similarity one, using only strict equivalences. The problem is that using strict equivalence misses some generalizations between rules. Consider, for example, the one occurrence of the correct rule in (18).

(18) \([NP \ the/DT 100/CD largest/JJS Nasdaq/NNP financial/JJ stocks/NNS \])

While DT CD JJS NNP JJ NNS may be the only daughters list of its kind, i.e., it has no equivalents, we can infer its correctness from similar rules in the treebank. For example, there are three instances of \( NP \rightarrow DT \ CD \ JJ \ NNP \ NNS \) in the WSJ, which are not strictly equivalent, but which are similar.

Thus, the search for dissimilar rules can be broadened, and this is what is done in Dickinson (2008). Namely, one can search for similar rules by using edit distance between rules; the principle put forth here that rules with few similar rules are the most anomalous is still true. Putting the idea of dissimilarity more directly in the task, one can also examine anomalous subsequences of daughters (e.g., bigrams) to find those which are the most problematic.

5.2 Portability

We here also sketch out ways in which the notion of similarity can be broadened to include richer treebank formalisms. The method we have used is based on a simple notion of rule similarity; thus, to apply it to treebanks with relatively flat context-free rules, one need only write a small set of treebank-specific adjunct categories and category mappings. The method faces challenges in extending to other types of treebanks, however, and we outline those challenges here, offering some pointers to solutions.

First, there are treebanks which contain discontinuous constituents, that is, constituents which are not contiguous. In example (19) from the TIGER treebank (Brants et al., 2002),
for instance, the noun phrase *Ein Mann det lacht* (‘a man who laughs’) is a complete constituent with material intervening.

(19) **Ein Mann** kommt, **der lacht**

    a man comes, who laughs

‘A man who laughs comes.’

We have assumed contiguous context-free rules in our definitions of rule similarity and dissimilarity, as categories are strictly ordered with respect to each other. In this new situation, we need a way to map discontinuous trees to context-free rules. Following work on context-free parsing with discontinuous treebanks (cf. Boyd, 2007), one possible solution is to split each constituent into its component parts. For example, with a split NP, we might have $S \rightarrow NP-d\,\text{VMFIN, NP-d}$ as a rule, where $-d$ marks one part of a discontinuous element. This would allow the NP elements to be compared with other NPs, while at the same time not making any assumptions about free word order.

Secondly, there are treebanks, which employ less flat structures, often only binary-branching trees (e.g., CCGbank (Hockenmaier and Steedman, 2007)). Obviously, this results in much fewer rule possibilities. The question for our method is: what information still highlights inconsistency?

Consider what happens when flat trees are binarized. In (20), for instance, we have an endocentricity violation. Here, it does not matter whether we have binary-branching or flatter trees; the fact that VP is the mother of POS is generally sufficient to indicate a problem. That is to say, endocentricity violations are in principle detectable with binary trees.

(20) a. It $[VP\,\text{'s/POS}\,\text{[NP ...]}\,\text{[SBAR ...]}\,\text{]}$]
   b. It $[VP\,\text{'s/POS}\,\text{[NP}\,\text{[NP ...]}\,\text{[SBAR ...]}\,\text{]}\,\text{]}$

Consider now how we would detect the overapplication of a flat structure, as shown in (21). To detect that RB JJ in (21b) should be ADJP, we either need to move beyond local trees, or we somehow need to flatten them. It seems that we need to examine trees which contain only maximal projections (e.g., NP in (21b)) and pre-terminal categories (e.g., RB, JJ, NN), skipping over intermediate levels.

(21) a. $[NP\,\text{a/DT\,fairly/RB\,systematic/JJ\,effort/NN ... ]}$
   b. $[NP\,\text{a/DT}\,\text{[N',\,fairly/RB\,[N',\,systematic/JJ\,\text{[N',\,effort/NN\,\text{[SBAR ...]}\,\text{]]]}]}\,\text{]}$

Finally, we can consider richer linguistic annotation, such as treebanks with head-driven phrase grammar (HPSG) and lexical-functional grammar (LFG) annotation, such as the kind used to convert treebanks into more deeply annotated resources (e.g., Miyao et al., 2004; O’Donovan et al., 2005). As the conversion process typically involves handwritten grammatical constraints, these treebanks can be expected to have a greater amount of consistency.

But the question remains: since nonterminal elements are now complex bundles of features instead of a simple category, how does one compare rules in this context? For our methods, the main questions to answer are: which features of the daughters predict the mother’s category, and which sequences can be seen to be anomalous? For
example, in an HPSG feature structure, one would likely want to use the value of the SYNSEM|LOC|CAT|HEAD path to determine the POS or category of each unit.

In this context, we can note that a potential benefit of our method is that it can assist in preprocessing of treebanks. That is, we can detect structures which lead to a failure of “grammar acquisition” (Miyao et al., 2004), and this requires very little work in extracting problematic constructions.

6 Summary and Outlook

We have shown how one can use basic linguistic abstractions to group treebank rules into equivalence classes. These equivalence classes of rules can be used, in the first place, to assist in detecting non-endocentric structures, by finding classes of rules which should have the same mother category, and, in the second place, to detect anomalous structures by finding rules which do not fit into an equivalence class with any other rules. This is done by using only a few simple, highly precise properties that maintain the predictive information of the daughters.

There are a variety of ways in which this work can be taken in the future. First, to validate the methods further, it needs to be applied to other treebanks and languages. Elucidating which properties of a treebank are predictive can provide further insights into general grammar compaction methods. In particular, applying equivalence mappings to treebanks with discontinuous and binary-branching representations (see section 5.2) will further the connections between flat context-free representations and other formalisms employing approximately the same amount of category information, but with different structural requirements.

Finally, one can use the insights of equivalence classes to explore regularities in the grammar. This could take the form of adding latent annotation (cf., e.g., Petrov et al., 2006) to subclasses of rules which have much in common, or this could involve generalizing the treebank grammar to predict new rules which did appear in the original data set.

Acknowledgments

I would like to thank Adriane Boyd and Detmar Meurers for their useful and insightful comments on earlier versions of this work, as well as the participants of the Fifth Workshop on Treebanks and Linguistic Theories for feedback on the rule similarity method. This material is based upon work supported by the National Science Foundation under Grant No. IIS-0623837.

References


Han, Chung-hye, Na-Rare Han, Eon-Suk Ko and Martha Palmer (2002). Development and Evaluation of a Korean Treebank and its Application to NLP. In Proceedings of LREC-02. Las Palmas, Canary Islands, Spain.


