High Precision Treebanking
— Blazing Useful Trees Using POS Information —

Takaaki Tanaka,[†] **Francis Bond,**[†] **Stephan Oepen,**[‡] **Sanae Fujita**[†]
[†] {takaaki, bond, fujita}@cslab.kecl.ntt.co.jp
[‡] oe@csli.stanford.edu
[†] NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation
[‡] Universitetet i Oslo and CSLI, Stanford

Abstract

In this paper we present a quantitative and qualitative analysis of annotation in the Hinoki treebank of Japanese, and investigate a method of speeding annotation by using part-of-speech tags. The Hinoki treebank is a Redwoods-style treebank of Japanese dictionary definition sentences. 5,000 sentences are annotated by three different annotators and the agreement evaluated. An average agreement of 65.4% was found using strict agreement, and 83.5% using labeled precision. Exploiting POS tags allowed the annotators to choose the best parse with 19.5% fewer decisions.

1 Introduction

It is important for an annotated corpus that the mark-up is both correct and, in cases where variant analyses could be considered correct, consistent. Considerable research in the field of word sense disambiguation has concentrated on showing that the annotation of word senses can be done correctly and consistently, with the normal measure being inter-annotator agreement (e.g. Kilgariff and Rosenzweig, 2000). Surprisingly, few such studies have been carried out for syntactic annotation, with the notable exceptions of Brants et al. (2003, p 82) for the German NeGra Corpus and Civit et al. (2003) for the Spanish Cast3LB corpus. Even such valuable and widely used corpora as the Penn TreeBank have not been verified in this way.

We are constructing the Hinoki treebank as part of a larger project in cognitive and computational linguistics ultimately aimed at natural language understanding (Bond et al., 2004). In order to build the initial syntactic and semantic models, we are treebanking the dictionary definition sentences of the most familiar 28,000 words of Japanese and building an ontology from the results.

Arguably the most common method in building a treebank still is manual annotation, annotators (often linguistics students) marking up linguistic properties of words and phrases. In some semi-automated treebank efforts, annotators are aided by POS taggers or phrase-level chunkers, which can propose mark-up for manual confirmation, revision, or extension. As computational grammars and parsers have increased in coverage and accuracy, an alternate approach has become feasible, in which utterances are parsed and the annotator selects the best parse Carter (1997); Oepen et al. (2002) from the full analyses derived by the grammar.

We adopted the latter approach. There were four main reasons. The first was that we wanted to develop a precise broad-coverage grammar in tandem with the treebank, as part of our research into natural language understanding. Treebanking the output of the parser allows us to immediately identify problems in the grammar, and improving the grammar directly improves the quality of the treebank in a mutually beneficial feedback loop (Oepen et al., 2004). The second reason is that we wanted to annotate to a high level of detail, marking not only dependency and constituent structure but also detailed semantic relations. By using a Japanese grammar (JACY: Siegel and Bender, 2002) based on a monostratal theory of grammar (HPSG: Pollard and Sag, 1994) we could simultaneously annotate syntactic and semantic structure without overburdening the annota-

Proceedings of the 43rd Annual Meeting of the ACL, pages 330–337,
Ann Arbor, June 2005.

tor. The third reason was that we expected the use of the grammar to aid in enforcing consistency — at the very least all sentences annotated are guaranteed to have well-formed parses. The flip side to this is that any sentences which the parser cannot parse remain unannotated, at least unless we were to fall back on full manual mark-up of their analyses. The final reason was that the discriminants can be used to update the treebank when the grammar changes, so that the treebank can be improved along with the grammar. This kind of dynamic, discriminant-based treebanking was pioneered in the Redwoods treebank of English (Oepen et al., 2002), so we refer to it as Redwoods-style treebanking.

In the next section, we give some more details about the Hinoki Treebank and the data used to evaluate the parser (§ 2). This is followed by a brief discussion of treebanking using discriminants (§ 3), and an extension to seed the treebanking using existing markup (§ 4). Finally we present the results of our evaluation (§ 5), followed by some discussion and outlines for future research.

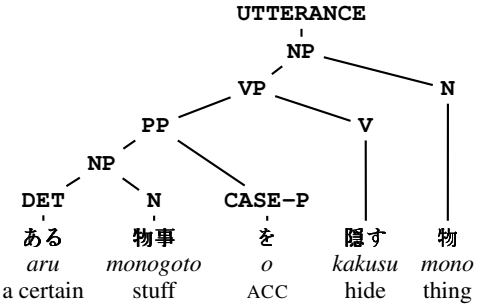
2 The Hinoki Treebank

The Hinoki treebank currently consists of around 95,000 annotated dictionary definition and example sentences. The dictionary is the Lexeed Semantic Database of Japanese (Kasahara et al., 2004), which consists of all words with a familiarity greater than or equal to five on a scale of one to seven. This gives 28,000 words, divided into 46,347 different senses. Each sense has a definition sentence and example sentence written using only these 28,000 familiar words (and some function words). Many senses have more than one sentence in the definition: there are 81,000 defining sentences in all.

The data used in our evaluation is taken from the first sentence of the definitions of all words with a familiarity greater than six (9,854 sentences). The Japanese grammar JACY was extended until the coverage was over 80% (Bond et al., 2004).

For evaluation of the treebanking we selected 5,000 of the sentences that could be parsed, and divided them into five 1,000 sentence sets (A–E). Definition sentences tend to vary widely in form depending on the part of speech of the word being defined — each set was constructed with roughly the same distribution of defined words, as well as having roughly the same length (the average was 9.9, ranging from 9.5–10.4).

A (simplified) example of an entry (Sense 2 of カーテン*kāten* "curtain: any barrier to communication or vision"), and a syntactic view of its parse are given in Figure 1. There were 6 parses for this definition sentence. The full parse is an HPSG sign, containing both syntactic and semantic information. A view of the semantic information is given in Figure 2[1].



Curtain$_2$: "a thing that hides something"

Figure 1: Syntactic View of the Definition of カーテン$_2$ *kāten* "curtain"

$$\begin{array}{ll}\langle h_0, x_2\ \{h_0 : proposition(h_5) & \\ \quad h_1 : aru(e_1, x_1, u_0) & \text{"a certain"} \\ \quad h_1 : monogoto(x_1) & \text{"stuff"} \\ \quad h_2 : u\_def(x_1, h_1, h_6) & \\ \quad h_5 : kakusu(e_2, x_2, x_1) & \text{"hide"} \\ \quad h_3 : mono(x_2) & \text{"thing"} \\ \quad h_4 : u\_def(x_2, h_3, h_7)\}\rangle & \end{array}$$

Figure 2: Semantic View of the Definition of カーテン$_2$ *kāten* "curtain"

The semantic view shows some ambiguity has been resolved that is not visible in the purely syntactic view. In Japanese, relative clauses can have gapped and non-gapped readings. In the gapped reading (selected here), 物 *mono* "thing" is the subject of 隠す *kakusu* "hide". In the non-gapped reading there is some unspecified relation between the thing and the verb phrase. This is similar to the difference in the two readings of *the day he knew* in English: "the day that he knew about" (gapped) vs "the day on which he knew (something)" (non-gapped).

[1]The semantic representation used is Minimal Recursion Semantics (Copestake et al., Forthcoming). The figure shown here hides some of the detail of the underspecified scope.

Such semantic ambiguity is resolved by selecting the correct derivation tree that includes the applied rules in building the tree, as shown in Figure 3. In the next phase of the Hinoki project, we are concentrating on acquiring an ontology from these semantic representations and using it to improve the parse selection (Bond et al., 2004).

3 Treebanking Using Discriminants

Selection among analyses in our set-up is done through a choice of *elementary discriminants*, basic and mostly independent contrasts between parses. These are (relatively) easy to judge by annotators. The system selects features that distinguish between different parses, and the annotator selects or rejects the features until only one parse is left. In a small number of cases, annotation may legitimately leave more than one parse active (see below). The system we used for treebanking was the [incr tsdb()] Redwoods environment[2] (Oepen et al., 2002). The number of decisions for each sentence is proportional to the log of the number of parses. The number of decisions required depends on the ambiguity of the parses and the length of the input. For Hinoki, on average, the number of decisions presented to the annotator was 27.5. However, the average number of decisions needed to disambiguate each sentence was only 2.6, plus an additional decision to accept or reject the selected parses[3]. In general, even a sentence with 100 parses requires only around 5 decisions and 1,000 parses only around 7 decisions. A graph of parse results versus number of decisions presented and required is given in Figure 6.

The primary data stored in the treebank is the derivation tree: the series of rules and lexical items the parser used to construct the parse. This, along with the grammar, can be combined to rebuild the complete HPSG sign. The annotators task is to select the appropriate derivation tree or trees. The possible derivation trees for カーテン$_2$ *kāten* "curtain" are shown in Figure 3. Nodes in the trees indicate applied rules, simplified lexical types or words. We will use it as an example to explain the annotation process. Figure 3 also displays POS tag from a separate tagger, shown in `typewriter font`.[4]

This example has two major sources of ambiguity. One is lexical: *aru* "a certain/have/be" is ambiguous between a reading as a determiner "a certain" (**det-lex**) and its use as a verb of possession "have" (**aru-verb-lex**). If it is a verb, this gives rise to further structural ambiguity in the relative clause, as discussed in Section 2. Reliable POS tags can thus resolve some ambiguity, although not all.

Overall, this five-word sentence has 6 parses. The annotator does not have to examine every tree but is instead presented with a range of 9 discriminants, as shown in Figure 4, each local to some segment of the utterance (word or phrase) and thus presenting a contrast that can be judged in isolation. Here the first column shows deduced status of discriminants (typically toggling one discriminant will rule out others), the second actual decisions, the third the discriminating rule or lexical type, the fourth the constituent spanned (with a marker showing segmentation of daughters, where it is unambiguous), and the fifth the parse trees which include the rule or lexical type.

D	*A*	Rules / Lexical Types	Subtrees / Lexical items	Parse Trees
?	?	**rel-cl-sbj-gap**	ある物事を隠す ‖ 物	2,4,6
?	?	**rel-clause**	ある物事を隠す ‖ 物	1,3,5
-	?	**rel-cl-sbj-gap**	ある ‖ 物事	3,4
-	?	**rel-clause**	ある ‖ 物事	5,6
+	?	**hd-specifier**	ある ‖ 物事	1,2
?	?	**subj-zpro**	隠す	2,4,6
-	?	**subj-zpro**	ある	5,6
-	?	**aru-verb-lex**	ある	3–6
+	+	**det-lex**	ある	1,2

+: positive decision
-: negative decision
?: indeterminate / unknown

Figure 4: Discriminants (marked after one is selected). *D* : deduced decisions, *A* : actual decisions

After selecting a discriminant, the system recalculates the discriminant set. Those discriminants which can be deduced to be incompatible with the decisions are marked with '−', and this information is recorded. The tool then presents to the annotator

[2]The [incr tsdb()] system, Japanese and English grammars and the Redwoods treebank of English are available from the Deep Linguistic Processing with HPSG Initiative (DELPH-IN: `http://www.delph-in.net/`).

[3]This average is over all sentences, even non-ambiguous ones, which only require a decision as to whether to accept or reject.

[4]The POS markers used in our experiment are from the ChaSen POS tag set (`http://chasen.aist-nara.ac.jp/`), we show simplified transliterated tag names.

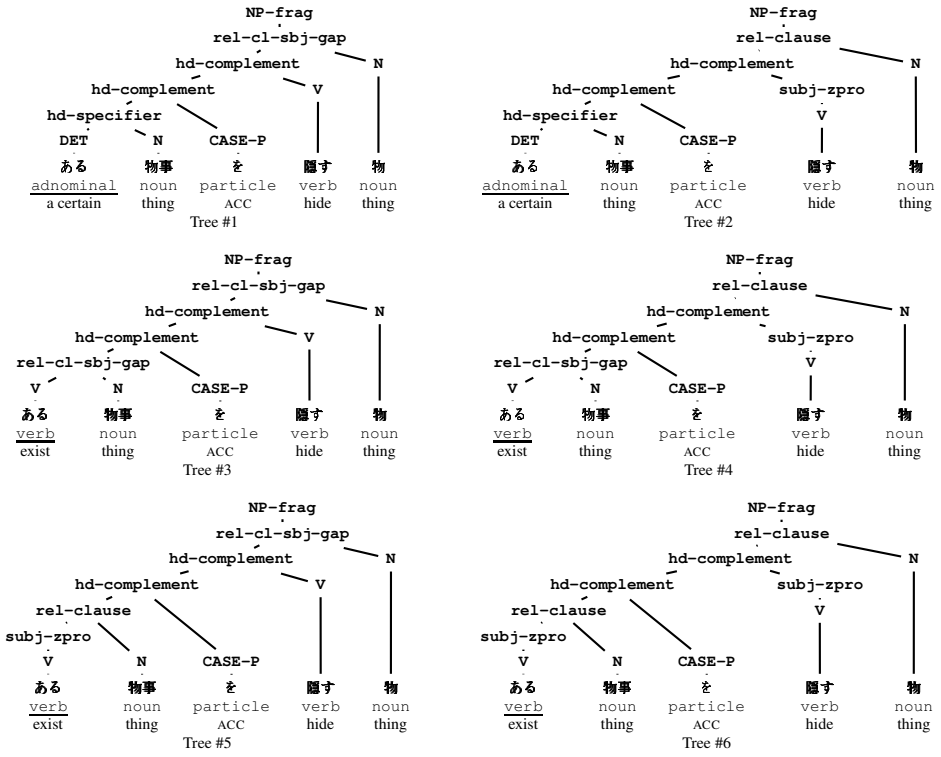


Figure 3: Derivation Trees of the Definition of カーテン$_2$ *kāten* "curtain"

only those discriminants which still select between the remaining parses, marked with '?'.

In this case the desired parse can be selected with a minimum of two decisions. If the first decision is that ある *aru* is a determiner (**det-lex**), it eliminates four parses, leaving only three discriminants (corresponding to trees #1 and #2 in Figure 3) to be decided on in the second round of decisions. Selecting 物 *mono* "thing" as the gapped subject of 隠す *kakusu* "hide" (**rel-cl-sbj-gap**) resolves the parse forest to the single correct derivation tree #1 in Figure 3.

The annotator also has the option of leaving some ambiguity in the treebank. For example, the verbal noun オープン *ōpun* "open" is defined with the single word 開く *aku/hiraku* "open". This word however, has two readings: *aku* which is intransitive and *hiraku* which is transitive. As オープン *ōpun* "open" can be either transitive or intransitive, both parses are in fact correct! In such cases, the annotators were instructed to leave both parses.

Finally, the annotator has the option of rejecting all the parses presented, if none have the correct syntax and semantics. This decision has to be made even for sentences with a unique parse.

4 Using POS Tags to Blaze the Trees

Sentences in the Lexeed dictionary were already part-of-speech tagged so we investigated exploiting this information to reduce the number of decisions the annotators had to make. More generally, there are many large corpora with a subset of the information we desire already available. For example, the Kyoto Corpus (Kurohashi and Nagao, 2003) has part of speech information and dependency information, but not the detailed information available from an HPSG analysis. However, the existing information can be used to blaze[5] trees in the parse forest: that is to select or reject certain discriminants based on existing information.

Because other sources of information may not be entirely reliable, or the granularity of the information may be different from the granularity in our

[5]In forestry, to blaze is to mark a tree, usually by painting and/or cutting the bark, indicating those to be cut or the course of a boundary, road, or trail.

treebank, we felt it was important that the blazes be defeasible. The annotator can always reject the blazed decisions and retag the sentence.

In [incr tsdb()], it is currently possible to blaze using POS information. The criteria for the blazing depend on both the grammar used to make the treebank and the POS tag set. The system matches the tagged POS against the grammar's lexical hierarchy, using a one-to-many mapping of parts of speech to types of the grammar and a subsumption-based comparison. It is thus possible to write very general rules. Blazes can be positive to accept a discriminant or negative to reject it. The blaze markers are defined to be a POS tag, and then a list of lexical types and a score. The polarity of the score determines whether to accept or reject. The numerical value allows the use of a threshold, so that only those markers whose absolute value is greater than a threshold will be used. The threshold is currently set to zero: all blaze markers are used.

Due to the nature of discriminants, having two positively marked but competing discriminants for the same word will result in no trees satisfying the conditions. Therefore, it is important that only negative discriminants should be used for more general lexical types.

Hinoki uses 13 blaze markers at present, a simplified representation of them is shown in Figure 5. E.g. if ⟨verb-aux, **v-stem-lex**, -1.0⟩ was a blaze marker, then any sentence with a verb that has two non-auxiliary entries (e.g. *hiraku/aku* vt and vi) would be eliminated. The blaze set was derived from a conservative inspection of around 1,000 trees from an earlier round of annotation of similar data, identifying high-frequency contrasts in lexical ambiguity that can be confidently blazed from the POS granularity available for Lexeed.

POS tags	Lexical Types in the Grammar	Score
verb-aux	**v-stem-lex**	−1.0
verb-main	**aspect-stem-lex**	−1.0
noun	**verb-stem-lex**	−1.0
adnominal	**noun_mod-lex-l**	0.9
	det-lex	0.9
conjunction	**n_conj-p-lex**	0.9
	v-coord-end-lex	0.9
adjectival-noun	**noun-lex**	−1.0

Figure 5: Some Blaze Markers used in Hinoki

For the example shown in Figures 3 and 4, the blaze markers use the POS tagging of the determiner ある *aru* to mark it as **det-lex**. This eliminates four parses and six discriminants leaving only three to be presented to the annotator. On average, marking blazes reduced the average number of blazes presented per sentence from 27.5 to 23.8 (a reduction of 15.6%). A graphical view of number of discriminants versus parse ambiguity is shown in Figure 6.

5 Measuring Inter-Annotator Agreement

Lacking a task-oriented evaluation scenario at this point, inter-annotator agreement is our core measure of annotation consistency in Hinoki. All trees (and associated semantics) in Hinoki are derived from a computational grammar and thus should be expected to demonstrate a basic degree of internal consistency. On the other hand, the use of the grammar exposes large amounts of ambiguity to annotators that might otherwise go unnoticed. It is therefore not *a priori* clear whether the Redwoods-style approach to treebank construction as a general methodology results in a high degree of internal consistency or a comparatively low one.

	$\alpha-\beta$	$\beta-\gamma$	$\gamma-\alpha$	**Average**
Parse Agreement	63.9	68.2	64.2	65.4
Reject Agreement	4.8	3.0	4.1	4.0
Parse Disagreement	17.5	19.2	17.9	18.2
Reject Disagreement	13.7	9.5	13.8	12.4

Table 1: Exact Match Inter-annotator Agreement

Table 1 quantifies inter-annotator agreement in terms of the harshest possible measure, the proportion of sentences for which two annotators selected the exact same parse or both decided to reject all available parses. Each set was annotated by three annotators (α, β, γ). They were all native speakers of Japanese with a high score in a Japanese proficiency test (Amano and Kondo, 1998) but no linguistic training. The average annotation speed was 50 sentences an hour.

In around 19 per cent of the cases annotators chose to not fully disambiguate, keeping two or even three active parses; for these we scored $\frac{i}{j}$, with j being the number of identical pairs in the cross-product of active parses, and i the number of mismatches. One annotator keeping $\{1, 2, 3\}$, for example, and another $\{3, 4\}$ would be scored as $\frac{1}{6}$. In addition to

leaving residual ambiguity, annotators opted to reject all available parses in some eight per cent of cases, usually indicating opportunities for improvement of the underlying grammar. The Parse Agreement figures (65.4%) in Table 1 are those sentences where both annotators chose one or more parses, and they showed non-zero agreement. This figure is substantially above the published figure of 52% for NeGra Brants et al. (2003). Parse Disagreement is where both chose parses, but there was no agreement. Reject Agreement shows the proportion of sentences for which both annotators found no suitable analysis. Finally Reject Disagreement is those cases were one annotator found no suitable parses, but one selected one or more.

The striking contrast between the comparatively high exact match ratios (over a random choice baseline of below seven per cent; $\kappa = 0.628$) and the low agreement between annotators on which structures to reject completely suggests that the latter type of decision requires better guidelines, ideally tests that can be operationalized.

To obtain both a more fine-grained measure and also be able to compare to related work, we computed a labeled precision f-score over derivation trees. Note that our inventory of labels is large, as they correspond in granularity to structures of the grammar: close to 1,000 lexical and 120 phrase types. As there is no 'gold' standard in contrasting two annotations, our labeled constituent measure F is the harmonic mean of standard labeled precision P (Black et al., 1991; Civit et al., 2003) applied in both 'directions': for a pair of annotators α and β, F is defined as:

$$F = \frac{2P(\alpha,\beta)P(\beta,\alpha)}{P(\alpha,\beta) + P(\beta,\alpha)}$$

As found in the discussion of exact match inter-annotator agreement over the entire treebank, there are two fundamentally distinct types of decisions made by annotators, viz. (a) elimination of unwanted ambiguity and (b) the choice of keeping at least one analysis or rejecting the entire item. Of these, only (b) applies to items that are assigned only one parse by the grammar, hence we omit unambiguous items from our labeled precision measures (a little more than twenty per cent of the total) to exclude trivial agreement from the comparison. In the same spirit, to eliminate noise hidden in pairs of items where one or both annotators opted for multiple valid parses, we further reduced the comparison set to those pairs where both annotators opted for exactly one active parse. Intersecting both conditions for pairs of annotators leaves us with subsets of around 2,500 sentences each, for which we record F values ranging from 95.1 to 97.4, see Table 2. When broken down by pairs of annotators and sets of 1,000 items each, which have been annotated in strict sequential order, F scores in Table 2 confirm that: (a) inter-annotator agreement is stable, all three annotators appear to have performed equally (well); (b) with growing experience, there is a slight increase in F scores over time, particularly when taking into account that set E exhibits a noticeably higher average ambiguity rate (1208 parses per item) than set D (820 average parses); and (c) Hinoki inter-annotator agreement compares favorably to results reported for the German NeGra (Brants, 2000) and Spanish Cast3LB (Civit et al., 2003) treebanks, both of which used manual mark-up seeded from automated POS tagging and chunking.

Compared to the 92.43 per cent labeled F score reported by Brants (2000), Hinoki achieves an 'error' (i.e. disagreement) rate of less than half, even though our structures are richer in information and should probably be contrasted with the 'edge label' F score for NeGra, which is 88.53 per cent. At the same time, it is unknown to what extent results are influenced by differences in text genre, i.e. average sentence length of our dictionary definitions is noticeably shorter than for the NeGra newspaper corpus. In addition, our measure is computed only over a subset of the corpus (those trees that can be parsed and that had multiple parses which were not rejected). If we recalculate over all 5,000 sentences, including rejected sentences (F measure of 0) and those with no ambiguity (F measure of 1) then the average F measure is 83.5, slightly worse than the score for NeGra. However, the annotation process itself identifies which the problematic sentences are, and how to improve the agreement: improve the grammar so that fewer sentences need to be rejected and then update the annotation. The Hinoki treebank is, by design, dynamic, so we expect to continue to improve the grammar and annotation continuously over the project's lifetime.

Test Set	$\alpha - \beta$ #	$\alpha - \beta$ F	$\beta - \gamma$ #	$\beta - \gamma$ F	$\gamma - \alpha$ #	$\gamma - \alpha$ F	Average F
A	507	96.03	516	96.22	481	96.24	96.19
B	505	96.79	551	96.40	511	96.57	96.58
C	489	95.82	517	95.15	477	95.42	95.46
D	454	96.83	477	96.86	447	97.40	97.06
E	480	95.15	497	96.81	484	96.57	96.51
	2435	96.32	2558	96.28	2400	96.47	96.36

Table 2: Inter-Annotator Agreement as Mutual Labeled Precision F-Score

Test Set	Annotator Decisions α	β	γ	Blazed Decisions
A	2,659	2,606	3,045	416
B	2,848	2,939	2,253	451
C	1,930	2,487	2,882	468
D	2,254	2,157	2,347	397
E	1,769	2,278	1,811	412

Table 3: Number of Decisions Required

5.1 The Effects of Blazing

Table 3 shows the number of decisions per annotator, including revisions, and the number of decisions that can be done automatically by the part-of-speech blazed markers. The test sets where the annotators used the blazes are shown underlined. The final decision to accept or reject the parses was not included, as it must be made for every sentence.

The blazed test sets require far fewer annotator decisions. In order to evaluate the effect of the blazes, we compared the average number of decisions per sentence for the test sets in which some annotators used blazes and some did not (B–D). The average number of decisions went from 2.63 to 2.11, a substantial reduction of 19.5%. similarly, the time required to annotate an utterance was reduced from 83 seconds per sentence to 70, a speed up of 15.7%. We did not include A and E, as there was variation in difficulty between test sets, and it is well known that annotators improve (at least in speed of annotation) over time. Research on other projects has shown that it is normal for learning curve differences to swamp differences in tools (Wallis, 2003, p. 65). The number of decisions against the number of parses is show in Figure 6, both with and without the blazes.

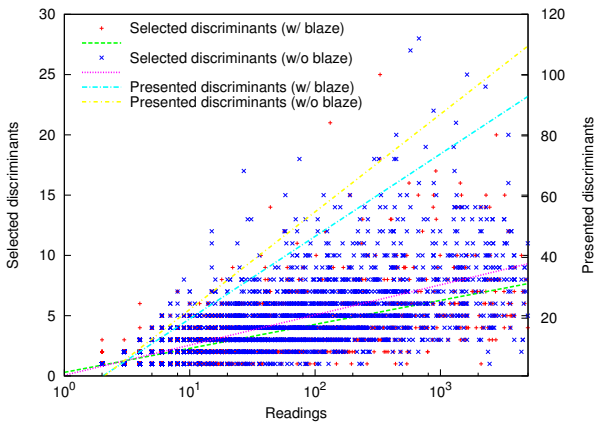


Figure 6: Number of Decisions versus Number of Parses (Test Sets B–D)

6 Discussion

Annotators found the rejections the most time consuming. If a parse was eliminated, they often redid the decision process several times to be sure they had not eliminated the correct parse in error, which was very time consuming. This shows that the most important consideration for the success of treebanking in this manner is the quality of the grammar. Fortunately, treebanking offers direct feedback to the grammar developers. Rejected sentences identify which areas need to be improved, and because the treebank is dynamic, it can be improved when we improve the analyses in the grammar. This is a notable improvement over semi-automatically constructed grammars, such as the Penn Treebank, where many inconsistencies remain (around 4,500 types estimated by Dickinson and Meurers, 2003) and the treebank does not allow them to be identified automatically or easily updated.

Because we are simultaneously using the semantic output of the grammar in building an ontology, and the syntax and semantics are tightly coupled, the knowledge acquisition provides a further route for feedback. Extracting an ontology from the semantic representations revealed many issues with the semantics that had previously been neglected.

Our top priority for further work within Hinoki

is to improve the grammar so as to both increase the cover and decrease the number of results with no acceptable parses. This will allow us to treebank a higher proportion of sentences, with even higher precision.

For more general work on treebank construction, we would like to investigate (1) using other information for blazes (syntactic constituents, dependencies, translation data) and marking blazes automatically using confident scores from existing POS taggers or parsers, (2) other agreement measures (for example agreement over the semantic representations), (3) presenting discriminants based on the semantic representations.

7 Conclusions

We conducted an experiment to measure inter-annotator agreement for the Hinoki corpus. Three annotators marked up 5,000 sentences. Sentence agreement was an unparalleled 65.4%. The method used identifies problematic annotations as a by-product, and allows the treebank to be improved as its underlying grammar improves. We also presented a method to speed up the annotation by exploiting existing part-of-speech tags. This led to a decrease in the number of annotation decisions of 19.5%.

Acknowledgments

The authors would like to thank the other members of the NTT Machine Translation Research Group, as well as Timothy Baldwin and Dan Flickinger. This research was supported by the research collaboration between the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University.

References

Anne Abeillé, editor. *Treebanks: Building and Using Parsed Corpora*. Kluwer Academic Publishers, 2003.

Shigeaki Amano and Tadahisa Kondo. Estimation of mental lexicon size with word familiarity database. In *International Conference on Spoken Language Processing*, volume 5, pages 2119–2122, 1998.

Ezra Black, Steven Abney, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Fred Jelinek, Judith Klavans, Mark Lieberman, and Tomek Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English. In *Proceedings of the Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, CA, 1991. Morgan Kaufmann.

Francis Bond, Sanae Fujita, Chikara Hashimoto, Kaname Kasahara, Shigeko Nariyama, Eric Nichols, Akira Ohtani, Takaaki Tanaka, and Shigeaki Amano. The Hinoki treebank: A treebank for text understanding. In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, pages 554–559, Hainan Island, 2004.

Thorsten Brants. Inter-annotator agreement for a German newspaper corpus. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, Athens, Greece, 2000.

Thorsten Brants, Wojciech Skut, and Hans Uszkoreit. Syntactic annotation of a German newspaper corpus. In Abeillé (2003), chapter 5, pages 73–88.

David Carter. The TreeBanker: a tool for supervised training of parsed corpora. In *ACL Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, Madrid, 1997. (`http://xxx.lanl.gov/abs/cmp-lg/9705008`).

Montserrat Civit, Alicia Ageno, Borja Navarro, Núria Bufi, and Maria Antonia Martí. Qualitative and quantitative analysis of annotators' agreement in the development of Cast3LB. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, Växjö, Sweeden, 2003.

Ann Copestake, Daniel P. Flickinger, Carl Pollard, and Ivan A. Sag. Minimal Recursion Semantics. An introduction. *Journal of Research in Language and Computation*, Forthcoming.

Markus Dickinson and W. Detmar Meurers. Detecting inconsistencies in treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, Växjö, Sweeden, 2003.

Kaname Kasahara, Hiroshi Sato, Francis Bond, Takaaki Tanaka, Sanae Fujita, Tomoko Kanasugi, and Shigeaki Amano. Construction of a Japanese semantic lexicon: Lexeed. SIG NLC-159, IPSJ, Tokyo, 2004. (in Japanese).

Adam Kilgariff and Joseph Rosenzweig. Framework and results for English SENSEVAL. *Computers and the Humanities*, 34 (1–2):15–48, 2000. Special Issue on SENSEVAL.

Sadao Kurohashi and Makoto Nagao. Building a Japanese parsed corpus — while improving the parsing system. In Abeillé (2003), chapter 14, pages 249–260.

Stephan Oepen, Dan Flickinger, and Francis Bond. Towards holistic grammar engineering and testing — grafting treebank maintenance into the grammar revision cycle. In *Beyond Shallow Analyses — Formalisms and Statistical Modeling for Deep Analysis (Workshop at IJCNLP-2004)*, Hainan Island, 2004. (`http://www-tsujii.is.s.u-tokyo.ac.jp/bsa/`).

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christoper D. Manning, Dan Flickinger, and Thorsten Brant. The LinGO redwoods treebank: Motivation and preliminary applications. In *19th International Conference on Computational Linguistics: COLING-2002*, pages 1253–7, Taipei, Taiwan, 2002.

Carl Pollard and Ivan A. Sag. *Head Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, 1994.

Melanie Siegel and Emily M. Bender. Efficient deep processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization at the 19th International Conference on Computational Linguistics*, Taipei, 2002.

Sean Wallis. Completing parsed corpora: From correction to evolution. In Abeillé (2003), chapter 4, pages 61–71.