High Precision Treebanking
– Blazing Useful Trees Using POS Information

Takaaki Tanaka[†] **Francis Bond**[†] **Stephan Oepen**[‡] **Sanae Fujita**[†]
[†] {bond,takaaki,fujita}@cslab.kecl.ntt.co.jp, [‡] oe@csli.stanford.edu
[†] NTT Communication Science Laboratories [‡] Universitet 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 were 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. Surprisingly, almost no 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 psycho-linguistics 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.

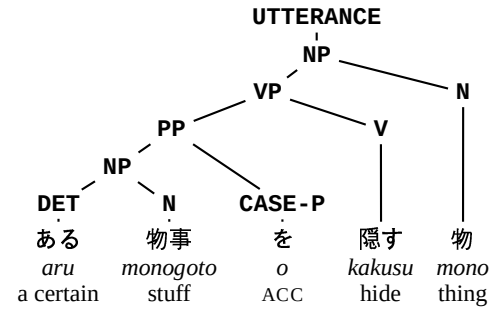
We adopted a semi-automated method in building a treebank, where annotators are aided by POS taggers or parsers. There were three main reasons. The first was that we wanted to develop a precise broad-coverage grammar in tandem with the treebank, as part of our research into natural language understanding. 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. 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 annotator. The third reason was that we expected the use of the grammar as a base to aid in enforcing consistency.

2 The Hinoki Treebank

The Hinoki treebank currently consists of 16,000 annotated dictionary definition 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.

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 is given in Figure 1(a). The full parse is an HPSG sign, containing both syntactic and semantic information. A view of the semantic information is given in Figure 1(b)[1].



(a) Syntactic View

$\langle h_0, x_2\{h_0 : prpstn\_rel(h_5)$
$h_1 : aru(e_1, x_1, u_0)$ "a certain"
$h_1 : monogoto(x_1)$ "stuff"
$h_2 : u\_def(x_1, h_1, h_6)$
$h_5 : kakusu(e_2, x_2, x_1)$ "hide"
$h_3 : mono(x_2)$ "thing"
$h_4 : u\_def(x_2, h_3, h_7)\}\rangle$

(b) Semantic View
Curtain$_2$: "a thing that hides something"

Figure 1: The Definition of カーテン$_2$ *kāten* "curtain"

[1]The semantic representation used is Minimal Recursion Semantics (Copestake et al., 1999). The figure shown here hides some of the detail of the underspecified scope.

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. The system we used for treebanking was the [incr tsdb()] profiling environment (Oepen and Carroll, 2000). 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 parser 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 the sentences was only 2.6, plus an additional decision to accept or reject the selected parses[2]. In general, even a sentence with 100 parses requires only around 5 decisions and 1,000 parses only around 7 decisions (Figure 4).

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. Nodes in the trees indicate applied rules, simplified lexical types or words. Each symbol below word is POS from a tagger output. We will use it as an example to explain the annotation process.

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 further structural ambiguity. In addition to the different parses arising from the different parts of speech, there is further ambiguity in the relative clause (gapped or non-gapped).These trees are divided into two groups, which can be discriminated only by simple POS produced by a tagger. Parser outputs both readings because it only use word segmentation from a tagger and does not use POS information. Relaible POS tags can reduce the number of trees, as described in the next section.

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 2, 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 having the rule or lexical type.

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 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 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.

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

4 Using POS Information to Blaze the Trees

Lexeed is 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. This can be used to blaze 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 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 criterion for the blazing depend on both the grammar used to make the treebank and the POS set. They are therefore kept in a separate file. The system matches the tagged POS against the grammars lexical hierarchy, using a one-to-many mapping of parts of speech to types of the grammar combined with 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 the accept/reject value. 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.

Hinoki uses 13 blaze markers at present, a simplified representation of them are shown in Figure 3[3]. 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. granularity available for Lexeed.

For the example shown in Figure 2, 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%). The reduction in the number of discriminants presented is shown in Figure 4.

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

[3]The actual POS markers used are from the ChaSen POS tagset (`http://chasen.aist-nara.ac.jp`).

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-specifi er	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 2: Discriminants (marked after one is selected). *D* : deduced decisions, *A* : actual decisions

(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)

Figure 3: Some Blaze Markers used in Hinoki

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.

	$\alpha - \beta$	$\beta - \gamma$	$\gamma - \alpha$	**Av.**
Parse Agreement	63.9	68.2	64.2	65.4
Parse Disagreement	17.5	19.2	17.9	18.2
Reject Agreement	4.8	3.0	4.1	4.0
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 without linguistic training. The average annotation speed was 50 sentences an hour.

The Parse Agreement figures (65.4%) in Table 1 are those sentences where both annotators chose one or more parses, and they showed some agreement. This figure is substantially above the published figure of 52% for NeGra 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 than can be operationalized.

To obtain both a more fine-grained measure and also be able to compare to related work, we computed a labeled

Test Set	$\alpha - \beta$ #	**F**	$\beta - \gamma$ #	**F**	$\gamma - \alpha$ #	**F**	**Av. 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

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 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). (c) Hinoki inter-annotator agreement compares favorably to results reported for NeGra and Cast3LB 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 If we recalculate over all 5,000 sentences, including rejected sentences and those with no ambiguity then the 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.

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.

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

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 test-sets B,C, and D, where some annotators used blazed sets and some did not. The average number of decisions went from 2.63 to 2.11, a substantial reduction of 19.5%. 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. The number of decisions against the number of parses is show in Figure 4, both with and without the blazes.

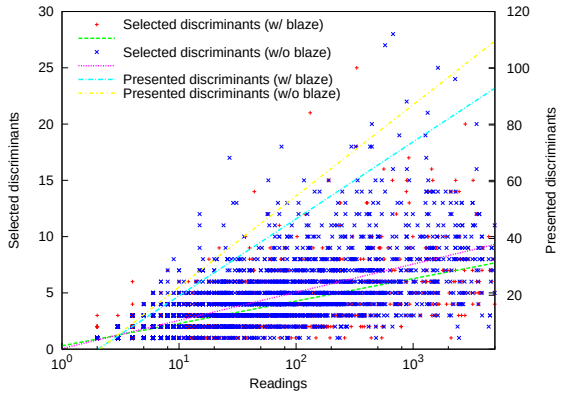


Figure 4: Number of Decisions versus Number of Parses

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 eliminate 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 that the treebank is dynamic, so it can go forward with 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.

7 Conclusions

We conducted an experiment to measure inter-annotator agreement for thew Hinoki corpus. Sentence agreement was an unparalleled 65.4%. This method allows the treebank to be improved as its underlying grammar improves. We also presented a method to speed up the tagging by exploiting existing part-of-speech tags. This led to a decrease in the number of annotation decisions of 19.5%.

References

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

Francis Bond, Sanae Fujita, Chikara Hashimoto, Kaname Kasahara, Shigeko Nariyama, Eric Nichols, Akira Ohtani, Takaaki Tanaka, and Shigeaki Amano. 2004. 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.

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

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

Montserrat Civit, Alicia Ageno, Borja Navarro, Núria Bufi , and Maria Antonia Martí. 2003. 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.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 1999. Minimal recursion semantics: An introduction. (manuscript `http://www-csli.stanford.edu/~aac/papers/newmrs.ps`).

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

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

Stephan Oepen and John Carroll. 2000. Performance profi l-ing for grammar engineering. *Natural Language Engineering*, 6(1):81–97.

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

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