The Hinoki Treebank: Working Toward Text Understanding

Francis Bond, Sanae Fujita, Chikara Hashimoto,[*]
Kaname Kasahara, Shigeko Nariyama,[†] Eric Nichols,[†]
Akira Ohtani,[‡] Takaaki Tanaka, Shigeaki Amano
NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation
[*]Kobe Shoin Women's University [†]NAIST [‡]Osaka Gakuin University
{bond, sanae, kaname, takaaki, amano}@cslab.kecl.ntt.co.jp [*] chashi@sils.shoin.ac.jp,
[†]{eric-n, shigeko}@is.naist.jp [‡]ohtani@utc.osaka-gu.ac.jp

Abstract

In this paper we describe the construction of a new Japanese lexical resource: the Hinoki treebank. The treebank is built from dictionary definition sentences, and uses an HPSG based Japanese grammar to encode the syntactic and semantic information. We show how this treebank can be used to extract thesaurus information from definition sentences in a language-neutral way using minimal recursion semantics.

1 Introduction

In this paper we describe the current state of a new lexical resource: the Hinoki treebank. The motivation and initial construction was described in detail in Bond et al. (2004a). The ultimate goal of our research is natural language understanding — we aim to create a system that can parse text into some useful semantic representation. Ideally this would be such that the output can be used to actually update our semantic models. This is an ambitious goal, and this paper does not present a completed solution, but rather a road-map to the solution, with some progress along the way.

The mid-term goal is to build a thesaurus from dictionary definition sentences and use it to enhance a stochastic parse ranking model that combines syntactic and semantic information. In order to do this the Hinoki project is combining syntactic annotation with word sense tagging. This will make it possible to test the use of similarity and/or class based approaches together with symbolic grammars and statistical models. Our aim in this is to alleviate data sparseness. In the Penn Wall Street Journal treebank (Taylor et al., 2003), for example, the words *stocks* and *skyrocket* never appear together. However, the superordinate concepts capital ($\supset$ *stocks*) and move upward ($\supset$ *skyrocket*) often do.

We are constructing the ontology from the machine readable dictionary Lexeed (Kasahara et al., 2004). This is a hand built self-contained lexicon: it consists of headwords and their definitions for the most familiar 28,000 words of Japanese. This set is large enough to include most basic level words and covers over 75% of the common word tokens in a sample of Japanese newspaper text. In order to make the system self sustaining we base the first growth of our treebank on the dictionary definition sentences themselves. We then train a statistical model on the treebank and parse the entire lexicon. From this we induce a thesaurus. We are currently tagging the definition sentences with senses. We will then use this information and the thesaurus to build a model that combines syntactic and semantic information. We will also produce a richer ontology — for example extracting selectional preferences. In the last phase, we will look at ways of extending our lexicon and ontology to less familiar words.

In this paper we present the results from treebanking 38,900 dictionary sentences. We also highlight two uses of the treebank: building the statistical models and inducing the thesaurus.

2 The Lexeed Semantic Database of Japanese

The Lexeed Semantic Database of Japanese consists of all Japanese words with a familiarity greater than or equal to five on a seven point scale (Kasahara et al., 2004). This gives 28,000 words in all, with 46,347 different senses. Definition sentences for these sentences were rewritten to use only the 28,000 familiar words (and some function words). The defining vocabulary is actually 16,900 different words (60% of all possible words). An example entry for first two senses of the word ドライバー *doraibā* "driver" is given in Figure 1, with English glosses added (underlined features are those added by Hinoki).

3 The Hinoki Treebank

The structure of our treebank is inspired by the Redwoods treebank of English in which utterances are parsed and the annotator selects the best parse from the full analyses derived by the grammar (Oepen et al., 2002). We had four main reasons for selecting this approach. 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 out-

INDEX	ドライバー *doraibā*	
POS	noun	Lexical-type noun-lex
FAMILIARITY	6.5 [1–7]	
SENSE 1	DEFINITION	ねじ/を/差し入れ/たり/ 、 抜き取っ/た/する/道具/。 “A tool for inserting and removing screws .”
	HYPERNYM	道具$_1$ *equipment* “tool”
	SEM. CLASS	⟨942:tool⟩ (⊂ 893:equipment)
SENSE 2	DEFINITION	自動車/を/運転/する/人/。 “Someone who drives a car .”
	HYPERNYM	人$_1$ *hito* “person”
	SEM. CLASS	⟨292:driver⟩ (⊂ 4:person)

Figure 1: Entry for the Word *doraibā* “driver” (with English glosses)

put 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 annotator. The treebank records the complete syntacto-semantic analysis provided by the HPSG grammar, along with an annotator's choice of the most appropriate parse. From this record, all kinds of information can be extracted at various levels of granularity. In particular, traditional syntactic structure (e.g., in the form of labeled trees), dependency relations between words and full meaning representations using minimal recursion semantics (MRS: Copestake et al. (1999)). A simplified example of the labeled tree, MRS and dependency views for the definition of ドライバー$_2$ *doraibā* “driver” is given in Figure 2.

The third reason was that we expect the use of the grammar as a base to aid in enforcing consistency — all sentences annotated are guaranteed to have well-formed parses. Experience with semi-automatically constructed grammars, such as the Penn Treebank, shows many inconsistencies remain (around 4,500 types estimated by Dickinson and Meurers (2003)) and the treebank does not allow them to be identified automatically.

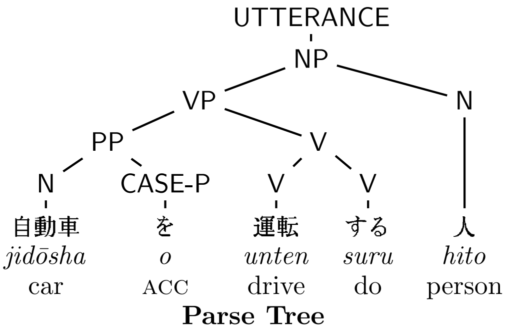
The last reason was the availability of a reasonably robust existing HPSG of Japanese (JACY), and a wide range of open source tools for developing the grammars. We made extensive use of the LKB (Copestake, 2002), a grammar development environment, in order to extend JACY to the domain of defining sentences. We also used the extremely efficient PET parser (Callmeier, 2000), which handles grammars developed using the LKB, to parse large test sets for regression testing, treebanking and finally knowledge acquisition. Most of our development was done within the [incr tsdb()] profiling environment (Oepen and Carroll, 2000). The existing resources enabled us to rapidly develop and test our approach.

3.1 Creating and Maintaining the Treebank

The construction of the treebank is a two stage process. First, the corpus is parsed (in our case using JACY with the PET parser), and then the annotator selects the correct analysis (or occasionally rejects all analyses). Selection is done through a choice of discriminants. The system selects features that distinguish between different parses, and the annotator selects or rejects the features until only one parse is left. The number of decisions for each sentence is proportional to $\log_2$ of the number of parses, although sometimes a single decision can reduce the number of remaining parses by more or less than half. In general, even a sentence with 5,000 parses only requires around 12 decisions.

Because the disambiguating choices made by the annotators are saved, it is possible to update the treebank when the grammar changes (Oepen et al., 2004). Although the trees depend on the grammar, re-annotation is only necessary in cases where either the parse has become more ambiguous, so new decisions have to be made, or existing rules or lexical items have changed so much that the system cannot reconstruct the parse.

One concern that has been raised with Redwoods style treebanking is the fact that the treebank is tied to a particular implementation of a grammar. The ability to update the treebank alleviates this concern to a large extent. A more serious concern is that it is only possible to annotate those trees that the grammar can parse. Sentences for which no analysis had been implemented in the grammar or which fail to parse due to processing constraints are left unannotated. This makes grammar coverage an urgent issue. However, dictionary definition sentences are more repetitive than newspaper text. In addition, there is little reference to outside context, and Lex-



$\langle h_0, x_1\{h_0 : proposition\_rel(h_1)$

$h_1 : hito(x_1)$ "person"

$h_2 : u\_def(x_1, h_1, h_6)$

$h_3 : jidosha(x_2)$ "car"

$h_4 : u\_def(x_2, h_3, h_7)$

$h_5 : unten(e_1, x_1, x_2)\}\rangle$ "drive"

MRS

$\{x_1 :$

$e_1 : unten(\text{ARG}_1\ x_1 : hito, \text{ARG}_2\ x_2 : jidosha)$

$r_1 : proposition(\text{MARG}\ e_1 : unten)\}$

Dependency

Figure 2: Parse Tree, Simplified MRS and Dependency Views for ドライバー$_2$ *doraibā* "driver"

eed has a fixed defining vocabulary. This makes it a relatively easy domain to work with.

We extended JACY by adding the defining vocabulary, and added some new rules and lexical-types (more detail is given in Bond et al. (2004a)).[1] Almost none of the rules are specific to the dictionary domain. The grammatical coverage over all sentences when we began to treebank was 84%, and it is currently being increased further as we work on the grammar. We have now treebanked all definition sentences for words with a familiarity greater than or equal to 6.0. This came to 38,900 sentences with an average length of 6.7 words/sentence. The extended JACY grammar is available for download from `www.dfki.uni-sb.de/~siegel/grammar-download/JACY-grammar.html`.

4 Applications

The treebanked data and grammar have been tested in two ways. The first is in training a stochastic model for parse selection. The second is in building a thesaurus from the parsed data.

4.1 Stochastic Parse Ranking

Using the treebanked data, we built a stochastic parse ranking model with [incr tsdb()]. The ranker uses a maximum entropy learner to train a PCFG over the parse derivation trees, with the current node as a conditioning feature. The correct parse is selected 61.7% of the time (training on 4,000 sentences and testing on another 1,000; evaluated per sentence). More feature-rich models using parent and grandparent nodes along with models trained on the MRS representations have been proposed and implemented with an English grammar and the Redwoods treebank (Oepen et al., 2002). We intend to include such features, as well as adding our own extensions to train on constituent weight and semantic class.

4.2 Knowledge Acquisition

We selected dictionary definitions as our first corpus in order to use them to acquire lexical and ontological knowledge. Currently we are classifying hypernym, hyponym, synonym and domain relationships in addition to linking senses to an existing ontology. Our approach is described in more detail in Bond et al. (2004b). The main difference between our research and earlier approaches, such as Tsurumaru et al. (1991), is that we are fully parsing the input, not just using regular expressions. Parsing sentences to a semantic representation (Minimal Recursion Semantics, Copestake et al. (1999)) has three advantages. The first is that it makes our knowledge acquisition somewhat language independent: if we have a parser for some language that can produce MRS, and a dictionary, the algorithm can easily be ported. The second reason is that we can go on to use the same system to acquire knowledge from non-dictionary sources, which will not be as regular as dictionaries and thus harder to parse using only regular expressions. Third, we can more easily acquire knowledge beyond simple hypernyms, for example, identifying synonyms through common definition patterns (Tsuchiya et al., 2001).

To extract hypernyms, we parse the first definition sentence for each sense. The parser uses the stochastic parse ranking model learned from the Hinoki treebank, and returns the MRS of the first ranked parse. Currently, 84% of the sentences can be parsed. In most cases, the word with the highest scope in the MRS representation will be the hypernym. For example, for *doraibā*$_1$ the hypernym is 道具 *dōgu* "tool" and for *doraibā*$_2$ the hypernym is 人 *hito* "person" (see Figure 1). Although the actual hypernym is in very different positions in the Japanese and English definition sentences, it takes the highest scope in both their semantic representations.

For some definition sentences (around 20%), further parsing of the semantic representation is necessary. For example, アナ$_1$ *ana* is defined as ***ana:***

[1] We benefited greatly from advice from the main JACY developers: Melanie Siegel and Emily Bender.

The abbreviation of "announcer" (translated to English). In this case *abbreviation* has the highest scope but is an explicit relation. We therefore parse to find its complement and extract the relationship abbreviation(ana$_1$,announcer$_1$). The semantic representation is largely language independent. In order to port the extraction to another language, we only have to know the semantic relation for abbreviation.

We evaluate the extracted pairs by comparison with an existing thesaurus: Goi-Taikei (Ikehara et al., 1997). Currently 58.5% of the pairs extracted for nouns are linked to nodes in the Goi-Taikei ontology (Bond et al., 2004b). In general, we are extracting pairs with more information than the Goi-Taikei hierarchy of 2,710 classes. In particular, many classes contain a mixture of class names and instance names: 豚肉 *buta niku* "pork" and 肉 *niku* "meat" are in the same class, as are ドラム *percussion instrument* "drum" and 打楽器 *dagakki* "percussion instrument", which we can now distinguish.

5 Conclusion and Further Work

In this paper we have described the current state of the Hinoki treebank. We have further showed how it is being used to develop a language-independent system for acquiring thesauruses from machine-readable dictionaries.

We are currently concentrating on three tasks. The first is improving the coverage of the grammar, so that we can parse more sentences to a correct parse. The second is improving the knowledge acquisition and learning other information from the parsed defining sentences — in particular lexical-types, semantic association scores, meronyms, and antonyms. The third task is adding the knowledge of hypernyms into the stochastic model.

With the improved the grammar and ontology, we will use the knowledge learned to extend our model to words not in Lexeed, using definition sentences from machine-readable dictionaries or where they appear within normal text. In this way, we can grow an extensible lexicon and thesaurus from Lexeed.

References

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

Francis Bond, Eric Nichols, Sanae Fujita, and Takaaki Tanaka. 2004b. Acquiring an ontology for a fundamental vocabulary. In *COLING 2004*, Geneva. (to appear).

Ulrich Callmeier. 2000. PET - a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(1):99–108.

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`).

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications.

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

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. 1997. *Goi-Taikei — A Japanese Lexicon*. Iwanami Shoten, Tokyo. 5 volumes/CDROM.

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 profiling for grammar engineering. *Natural Language Engineering*, 6(1):81–97.

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

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

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

Melanie Siegel and Emily M. Bender. 2002. Efficient 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.

Ann Taylor, Mitchel Marcus, and Beatrice Santorini. 2003. The Penn treebank: an overview. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, chapter 1, pages 5–22. Kluwer Academic Publishers.

Masatoshi Tsuchiya, Sadao Kurohashi, and Satoshi Sato. 2001. Discovery of definition patterns by compressing dictionary sentences. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium, NLPRS2001*, pages 411–418, Tokyo.

Hiroaki Tsurumaru, Katsunori Takesita, Itami Katsuki, Toshihide Yanagawa, and Sho Yoshida. 1991. An approach to thesaurus construction from Japanese language dictionary. In *IPSJ SIGNotes Natural Language*, volume 83-16, pages 121–128. (in Japanese).