

# The Hinoki Treebank

## A Treebank for 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 Telephone and Telegraph Corporation

\*Kobe Shoin Women's University †NAIST ‡Osaka Gakuin University

### Abstract

In this paper we describe the construction of a new Japanese lexical resource: the Hinoki treebank. The tree bank is built from dictionary definition sentences, and uses an HPSG grammar to encode the syntactic and semantic information. We then show how this treebank can be used to extract thesaurus information from definition sentences in a language-neutral way using Minimal Recursion Semantics. Finally we outline how this treebank can be expanded from its base.

### 1 Introduction

In this paper we describe the construction of a new lexical resource: the Hinoki treebank. We present the motivation for its construction, and a preliminary application. The ultimate goal of our research is natural language understanding - we aim to take text and parse it to 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.

Recently significant improvements have been made in combining symbolic and statistical approaches to various natural language processing tasks. For example, in parsing symbolic grammars are being combined with stochastic models (Toutanova et al., 2002). In question answering, se-

mantic tags are being combined with statistical models (Hori et al., 2003). Statistical techniques have also been shown to be useful for word sense disambiguation (Stevenson, 2003). However, to date, there have been no combinations of sense information together with symbolic grammars and statistical models. Klein and Manning (2003) show that much of the gain in statistical parsing using lexicalized models comes from the use of a small set of function words. General relations between words do not provide a lot of traction, presumably because the data is too sparse: in the Penn treebank normally used to train and test statistical parsers *stocks* and *skyrocket* never appear together. They note that this should motivate the use of similarity and/or class based approaches: the superordinate concepts capital ( $\supset$  *stocks*) and move upward ( $\supset$  *sky rocket*) frequently appear together. However, there has been little success in this area to date.

We feel there are two major reasons for the lack of progress. The first reason is that there simply is no single resource that combines syntactic and semantic tagging in a single corpus, so it is impossible to train statistical models using both sources of information. The second is that it is still not clear exactly what kind of semantic information is necessary or how to obtain it.

Our proposed solution to these problems has three phases. In the first phase, we are building a treebank using the Japanese semantic database Lexeed. This is a hand built self-contained lexicon: it consists of headwords and their definitions for the most familiar 28,000 words of Japanese, with all the definitions using only those 28,000 words (and some function

words). This set is large enough to include most basic level words and covers over 75% of the common words in a typical Japanese newspaper (counted by token) (Kanasugi et al., 2002). We then train a statistical model on the treebank and use it to help us induce a thesaurus. In phase two, we will sense tag the definition sentences and use this information and the thesaurus to build a model that combines syntactic and semantic information. We will also produce a richer ontology — at least extracting qualia structures and selectional preferences. In phase three, we will look at ways of extending our lexicon and ontology to less familiar words.

In this paper we discuss preliminary results from phase one. In particular, we introduce the construction of the treebank, building the statistical models and inducing the thesaurus. The technologies we are using in phase one are not new, the novelty is in the combination.

In the following section we give more information about Lexeed. Then, in Section 3 we discuss the creation of the treebank, Hinoki. The design is inspired by the Redwoods treebank of English (Oepen et al., 2002) a dynamic treebank closely linked to an HPSG analysis. It uses the JACY Japanese grammar (Siegel and Bender, 2002).

We next show how we can use lexeed corpus and the grammar used in the treebank to create a thesaurus (§ 4).

Finally, we outline in more detail our path to the goal of understanding Japanese (§ 6)

## 2 The Lexeed Semantic Database of Japanese

The Lexeed Semantic Database of Japanese aims to cover the most common words in Japanese. It was built based on a series of psycholinguistic experiments where words from two existing machine-readable dictionaries were presented to subjects and they were asked to rank them on a familiarity scale from one to seven, with seven being the most familiar (Kanasugi et al., 2002; Amano and Kondo, 1999).

Lexeed consists of all words with a familiarity greater than or equal to five. There are 28,000 words in all. Many words have multiple senses, there were 46,347 different senses. Definition sentences for

these sentences were rewritten by four different analysts to use only the 28,000 familiar words and the best definition chosen by a second set of analysts. In the final configuration, 18,700 different words (66% of all possible words) were actually used in the definition sentences. An example entry for the word 檜 *hinoki* “Japanese Cedar” is given in Figure 1, with English glosses added. It has a familiarity of above 5, and only one sense, which is defined with several defining sentences. In all there are 81,000 defining sentences. Our goal in phase one of our research is to create a treebank of these 81,000 sentences.

## 3 The Hinoki Treebank

The structure of our treebank is based the Redwoods treebank of English (Oepen et al., 2002) a dynamic treebank closely linked to an HPSG analysis. We chose this for several reasons. The most important is that the representation is very rich. The treebank records the complete syntacto-semantic analysis provided by an 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)).

Another important reason was the availability of a reasonably robust existing HPSG grammar of Japanese: JACY (Siegel and Bender, 2002); and a wide range of open source tools for developing the grammars used. 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). In addition to its well documented facilities for comparing different versions of a grammar (or the same grammar using different parsers), it has facilities for annotating treebanks, updating them and training stochastic models using them. These models can

DEFINITION WORD	檜 <i>hinoki</i> Japanese Cedar								
VARIANTS	桧, ヒノキ White Cedar								
FAMILIARITY	5.469 [1–7]								
DEFINITION	Sense 1								
	<table border="1"> <tr> <td>S1</td> <td>一年を通して葉が緑色の高い木。 tall <u>tree</u> whose leaves remain green at all times of the year</td> </tr> <tr> <td>S2</td> <td>高さが30メートル、直径1メートルに達する。 height reaches 30 meters, diameter 1 meter.</td> </tr> <tr> <td>...</td> <td></td> </tr> <tr> <td>S5</td> <td>材は優良な建築材。 lumber is an excellent building material</td> </tr> </table>	S1	一年を通して葉が緑色の高い木。 tall <u>tree</u> whose leaves remain green at all times of the year	S2	高さが30メートル、直径1メートルに達する。 height reaches 30 meters, diameter 1 meter.	...		S5	材は優良な建築材。 lumber is an excellent building material
S1	一年を通して葉が緑色の高い木。 tall <u>tree</u> whose leaves remain green at all times of the year								
S2	高さが30メートル、直径1メートルに達する。 height reaches 30 meters, diameter 1 meter.								
...									
S5	材は優良な建築材。 lumber is an excellent building material								

Figure 1: Entry for the word *hinoki* “Japanese Cedar” (with English glosses)

then be used by PET to selectively rank the parser output.

### 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 next. The number of decisions for each sentence is normally around  $\log_2$  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. An example of the selection tool is given in Figure 2.

Saving the disambiguating choices made by the annotators crucially allows the system to update the treebank when the grammar changes (Oepen et al., 2002). 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.

[incr() tsdb] stores all its information in

plain text files. This means that it is possible to make notational changes in the grammar (such as renaming all lexical items to use a more consistent naming scheme) and then apply the changes directly to the database with any tools that manipulate text. We did this once, simultaneously changing the names in both the grammar and the treebank with a simple perl script and the new treebank worked flawlessly.

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. We felt that the ability to update the treebank alleviates this concern to a large extent. Another 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. In the next section we discuss how we extended the grammar coverage in order to build the treebank.

### 3.2 Extending the Grammar

An inherent feature of deep grammars is that they are complicated. In practice this means that there is often one or two people who fully comprehends the grammar and is able to extend without an exorbitant investment in time needed to study the grammar. This is not a problem only of grammars, it is a

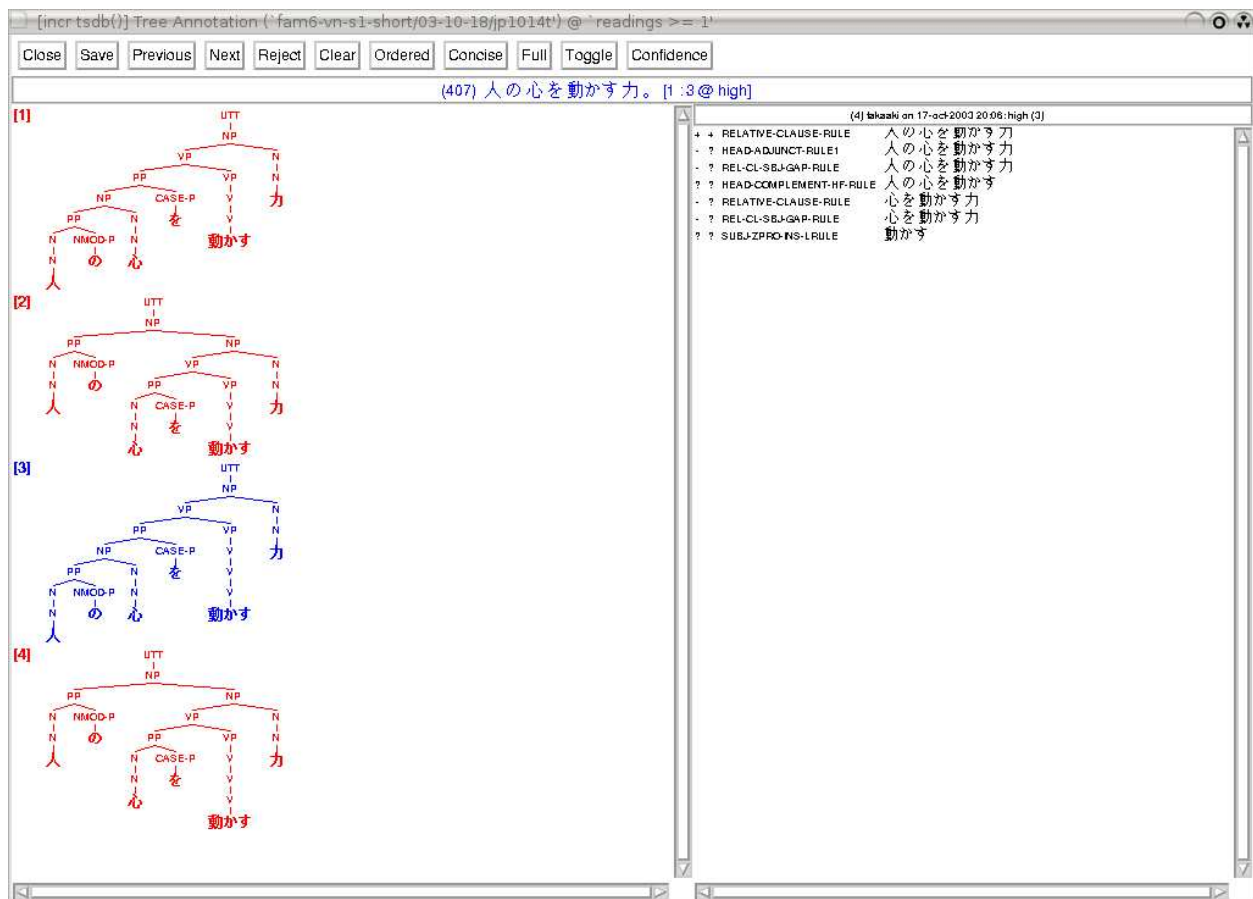


Figure 2: Parse selection

well known in software management.

Testing JACY on the full set of 81,000 defining sentences from Lexeed gave a coverage of 39.3%, using the inbuilt unknown word mechanism. This was trivially extended to 46.2% by adding some orthographic variants. The developers of JACY were successful in porting the grammar to new domains, with an increase of coverage of 48% in three months going from speech data to banking text, and then an increase of 51% in only two weeks going to a new domain of email requests (Siegel and Bender, 2002).

We decided to test JACY's usability by attempting to bring the cover on the Lexeed defining sentences to over 80% in 4 weeks. Six people were involved in this task, none of whom were involved in the original JACY development, and half of whom had little experience with HPSG. We expected it to be a relatively easy domain to port to, (Barnbrook, 2002) showed that for English defining sentences, some 10

sentence types covered over 80% of all entries.

As we also wanted to experiment on treebanking in the same time-frame, we restricted ourselves to considering only the first defining sentence for each sense of all words with a familiarity greater than or equal to 6.0. This came to some 10,000 sentences, with an average length of 10.1 words/sentence. Finally, because we wanted to enter full syntactic information for all of the words in Lexeed, we switched off the unknown word processing. This gave us an initial cover of around 10%.

We were able to bring the coverage to over 80% within the four weeks. The results are shown in Figures 3 and 4 which show the increase in coverage and ambiguity respectively.

The first big increase in coverage (to 55%) came from automatically expanding the lexicon. Tuning the lexicon and rules led to some incremental gains, mainly from relaxing the constraints on some exist-

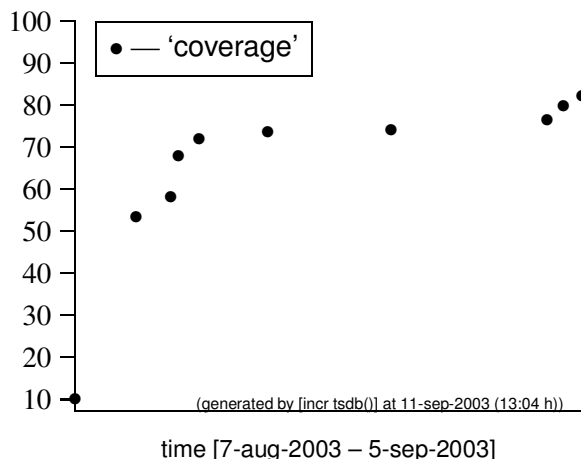


Figure 3: Evolution of coverage

ing rules. We also added some new rules<sup>1</sup>, for example a rule to parse verb-verb constructions. This bought us to over 70%, at which point we started treebanking. Up until this point, none of the rules we added were specific to the definition domain.

After two weeks treebanking, we added a new domain-specific rule for definitions such as *cat*: *In computing, a command that outputs a file to standard output*. In this case the clause *in computing* does not modify anything internal to the definition, but is effectively external to it *cat*: *In computing, [cat means] a command that . . .* To handle this, we added a construction which effectively adds a constructionally defined predicate above a noun phrase, if and only if there is something extra to modify the phrase and it is the highest constituent (the root). Although the implementation of the construction was specific to Japanese, the idea is not at all language specific, and the resulting semantic representation is language agnostic. We also added several other small improvements.

Keeping the ambiguity as low as possible was very important from the point of view of building the treebank. Every extra ambiguity meant more work in selecting the best parse. However, as coverage increased, unavoidably real ambiguity also increased. Occasionally, a rule would cause massive spurious ambiguity. In our first attempt to allow adverbial

<sup>1</sup>In which we benefitted greatly from some advice from the JACY developers.

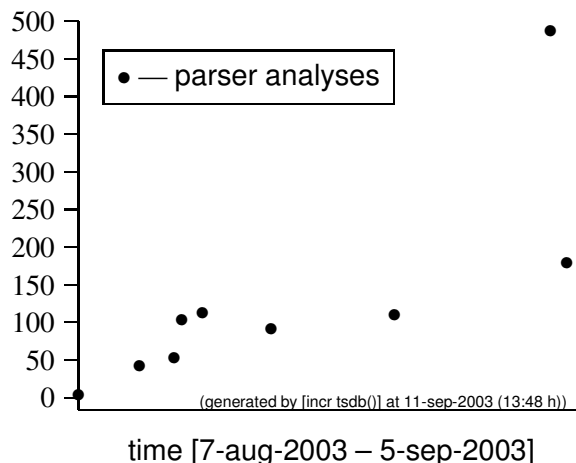


Figure 4: Evolution of Ambiguity

modification of root noun phrase fragments, we allowed adverbial modification of any noun phrase fragment, which sent the ambiguity sky-rocketing to around 500 parses per sentence. The final ambiguity is around 180 parses/sentence. This means that on average each tree requires 7-8 decisions to disambiguate it fully.

The extended JACY grammar is available for download at [www.delph-in.net/index.php3?page=3](http://www.delph-in.net/index.php3?page=3).<sup>2</sup>

### 3.3 Current Status

We have now treebanked around 5,000 of the 10,000 sentences with familiarity  $\geq 6$ . Looking at these sentences, 95% were able to be resolved to one correct parse. Around 5% had no correct parse, mainly due to two errors - one in the construction of the semantic representation for determiners (which has been fixed, but the treebank has not been updated) and one in the way coordinate constructions are constructed. The annotator could not settle on a single correct parse for fewer than 1% of the sentences.

Using the treebanked data, we built a simple maximum-entropy based stochastic parse ranking model using the tools supplied with `[incr() tsdb]`. Using this, training on 2,000 sentences and testing on another 1,000 the correct parse is se-

<sup>2</sup>At least it will be by the time the camera-ready paper is submitted. Looking inside the grammar completely blows our anonymity, so please don't peek until after you have finished reviewing.

lected 72% of the time (evaluated per sentence) an improvement of 53% over the baseline of random choice (19%). Training on only 1,000 sentences gave an accuracy of 58%. We are currently evaluating a larger test set.

#### 4 Knowledge Acquisition

In addition to our work on the development of a Japanese language HPSG treebank, we are performing knowledge acquisition on the corpus of dictionary definition sentences used in its construction. Our goals are to see what kind of hierarchical information about the entry words corresponding to the definition sentences can be obtained. It is also our intention to evaluate the potential for application of this information outside the field of lexicography, such as in machine translation or word sense disambiguation.

Past research in knowledge acquisition from definition sentences in Japanese has primarily dealt with the task of automatically generating hierarchy structures. Tsurumaru et al. (1991) developed a system for automatic thesaurus construction based on information derived from analysis of the terminal clauses of definition sentences. It was successful in classifying hyperonym, hyponym, and synonym relationships; however, it lacked any concrete evaluation of the accuracy of the hierarchies created. More recently (Tokunaga et al., 2001) created an ontology from a machine-readable dictionary and combined it with an existing thesaurus (Ikehara et al., 1997).

Our method differs from the aforementioned in three respects: first, prior research has been limited to nouns, where our method handles all parts of speech; second, our seed lexicon is self-contained — all the defining words are defined; third, we are fully parsing the input not just using regular expressions.

It is this third difference, the use of a well defined semantics (Minimal Recursion Semantics: Copestake et al. (1999)) that is the most important. There are three reasons: the first is that it makes our knowledge acquisition somewhat language independent: if we have a parser that can produce MRS, and a dictionary for that language, the algorithm can easily be ported. The second reason it is worth parsing, 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 just regular expressions. Third, we can more easily acquire knowledge beyond simple hypernyms, for example, identifying synonyms through common definition patterns as proposed by Tsuchiya et al. (2001).

We selected Minimal Recursion Semantics as our semantics because it generates flat semantic structures that are compatible across different languages. This flat semantics makes it very tractable for NLP applications. It is also produced by grammars for an increasingly wide number of languages, with the most developed being Japanese, English (ERG) and German (DISCO), and recent work producing grammars for Korean, Norwegian, Italian and Modern Greek.

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, 87% of sentences can be parsed. In most cases, the word with the highest scope in the MRS representation will be the hypernym. For example, for the word 檜 *hinoki*, the hypernym is 木 *ki* “tree” (see Figure 1). Although the actual hypernym is in very different positions in the two definition sentences, it takes the highest scope in either of their semantic representations.

For some definition sentences (around 20%), further parsing of the semantic representation is necessary. For example, given a definition sentence such as *cat*: *In computing, [cat means] a command that outputs a file to standard output*, the hypernym will be the complement of the constructionally defined predicate. Again, this semantic representation is not language dependent, we will not have to rebuild everything for a new language. Further, as we expand the scope of the knowledge acquisition the parsing can give us more information — this sense of *cat* is used in the domain of computing.

We are currently preparing a quantitative evaluation based on a comparison with an existing thesaurus: the Goi-Taikei (Ikehara et al., 1997). Some examples where the hypernym allows us to correctly link different senses to the correct nodes in the Goi-Taikei ontology are given in Table 1.

Our aim is to however to go beyond the Goi-

Word	Gloss	Hypernym	Gloss	Node
ドライバー	driver	人	person	worker
		道具	tool	tool
ドクター	doctor	博士	PhD	title
		医者	MD	medical worker

Table 1: Sense Disambiguation using Hypernyms

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”. This conflation has caused problems in applications such as question answering as well as fundamental research on linking syntax and semantics (Bond and Vatikiotis-Bateson, 2002).

In conclusion, our method is capable of thesaurus information in a principled manner. Our future work includes preparing a full quantitative evaluation, and implementing the means to acquire other entry word / definition word relationships, such as meronymy and synonymy.

## 5 Conclusion

In this paper we motivated the construction of a new lexical resource: the Hinoki treebank, and described its initial construction. We further showed how it can be used to develop a language independent system for acquiring thesauruses from machine-readable dictionaries.

## 6 Further Work

The first step in our path toward a full understanding of Japanese is to complete phase one by treebanking all the defining sentences in Lexeed. This means that we must also improve the coverage of the grammar, so that we can parse all sentences. When we have done that we will retrain our statistical model and use the new grammar to relearn the hypernym relations with a higher precision.

In phase two we will add the knowledge of hypernyms into the stochastic model, and look at learning other information from the parsed defining sentences — in particular semantic association scores and syntactic lexical-types.

In phase three, we will use the acquisition models learned in phase two, we can then go on 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 our fertile seed.

## References

- Shigeaki Amano and Tadahisa Kondo. 1999. *Nihongo-no Goi-Tokusei (Lexical properties of Japanese)*. Sanseido.
- Geoff Barnbrook. 2002. *Defining Language — A local grammar of definition sentences*. Studies in Corpus Linguistics. John Benjamins.
- Francis Bond and Caitlin Vatikiotis-Bateson. 2002. Using an ontology to determine English countability. In *19th International Conference on Computational Linguistics: COLING-2002*, volume 1, pages 99–105, Taipei.
- 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.
- Chiori Hori, Takaaki Hori, Hideki Isozaki, Eisaku Maeda, Shigeru Katagiri, and Sadaoki Furui. 2003. Deriving disambiguous queries in a spoken interactive ODQA system. In *ICASSP-2003*, pages 624–627.
- 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.

- Yuuko Kanasugi, Kaname Kasahara, Nozomi Inago, and Shigeaki Amano. 2002. Selection of a basic vocabulary based on word familiarity ratings. In *IEICE Technical Report NLC2002*, number 27, pages 21–26. IEICE. (in Japanese).
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430.
- Stephan Oepen and John Carroll. 2000. Performance profiling for grammar engineering. *Natural Language Engineering*, 6(1):81–97.
- Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christopher D. Manning. 2002. LinGO redwoods: A rich and dynamic treebank for HPSG. In *Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT2002)*, Sozopol, Bulgaria.
- Melanie Siegel and Emily M. Bender. 2002. 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.
- Mark Stevenson. 2003. *Word Sense Disambiguation*. CSLI Publications.
- Takenobu Tokunaga, Yasuhiro Syotu, Hozumi Tanaka, and Kiyooki Shirai. 2001. Integration of heterogeneous language resources: A monolingual dictionary and a thesaurus. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium, NLP-RS2001*, pages 135–142, Tokyo.
- Kristina Toutanova, Christopher D. Manning, and Stephan Oepen. 2002. Parse ranking for a rich HPSG grammar. In *Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT2002)*, Sozopol, Bulgaria.
- 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, NLP-RS2001*, 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).