# Corpus-Oriented Development of Japanese HPSG Parsers

**Kazuhiro Yoshida**
Department of Computer Science,
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033
kyoshida@is.s.u-tokyo.ac.jp

## Abstract

This paper reports the corpus-oriented development of a wide-coverage Japanese HPSG parser. We first created an HPSG treebank from the EDR corpus by using heuristic conversion rules, and then extracted lexical entries from the treebank. The grammar developed using this method attained wide coverage that could hardly be obtained by conventional manual development. We also trained a statistical parser for the grammar on the treebank, and evaluated the parser in terms of the accuracy of semantic-role identification and dependency analysis.

## 1 Introduction

In this study, we report the corpus-oriented development of a Japanese HPSG parser using the EDR Japanese corpus (2002). Although several researchers have attempted to utilize linguistic grammar theories, such as LFG (Bresnan and Kaplan, 1982), CCG (Steedman, 2001) and HPSG (Pollard and Sag, 1994), for parsing real-world texts, such attempts could hardly be successful, because manual development of wide-coverage linguistically motivated grammars involves years of labor-intensive effort.

Corpus-oriented grammar development is a grammar development method that has been proposed as a promising substitute for conventional manual development. In corpus-oriented methods, a treebank of a target grammar is constructed first, and various grammatical constraints are extracted from the treebank. Previous studies reported that wide-coverage grammars can be obtained at low cost by using this method. (Hockenmaier and Steedman, 2002; Miyao et al., 2004) The treebank can also be used for training statistical disambiguation models, and hence we can construct a statistical parser for the extracted grammar.

The corpus-oriented method enabled us to develop a Japanese HPSG parser with semantic information, whose coverage on real-world sentences is 95.3%. This high coverage allowed us to evaluate the parser in terms of the accuracy of dependency analysis on real-world texts, the evaluation measure that is previously used for more statistically-oriented parsers.

## 2 HPSG

Head-Driven Phrase Structure Grammar (HPSG) is classified into lexicalized grammars (Schabes et al., 1988). It attempts to model linguistic phenomena by interactions between a small number of grammar rules and a large number of lexical entries. Figure 1 shows an example of an HPSG derivation of a Japanese sentence ‘kare ga shinda,’ which means, ‘He died.’ In HPSG, linguistic entities such as words and phrases are represented by typed feature structures called *signs*, and the grammaticality of a sentence is verified by applying grammar rules to a sequence of signs. The sign of a lexical entry encodes the type and valence (i.e. restriction on the types of phrases that can appear around the word) of a corresponding word. Grammar rules of HPSG consist of



*Proceedings of the ACL Student Research Workshop*, pages 139–144,
Ann Arbor, Michigan, June 2005. 

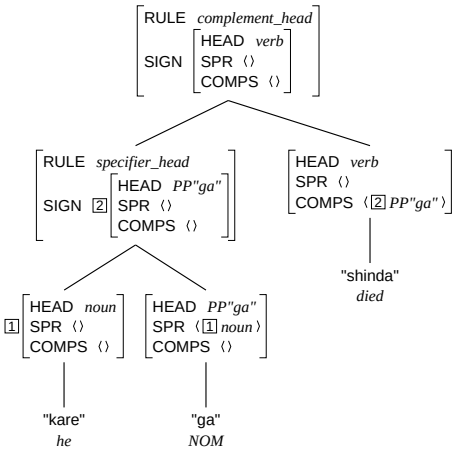


Figure 1: Example of HPSG analysis.

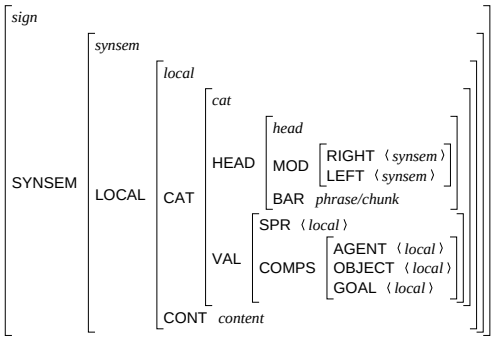


Figure 2: Sign of the grammar.

*schemata* and *principles*, the former enumerate possible patterns of phrase structures, and the latter are basically for controlling the inheritance of daughters' features to the parent.

In the current example, the lexical entry for "shinda" is of the type *verb*, as indicated in its *HEAD*, and its *COMPS* feature restricts its preceding phrase to be of the type *PP"ga"*. The *HEAD* feature of the root node of the derivation is inherited from the lexical entry for "shinda", because *complement-head* structures are head-final, and the *head feature* principle states that the *HEAD* feature of a phrase must be inherited from its head daughter.

There are several implementations of Japanese HPSG grammars. JACY (Siegel and Bender, 2002) is a hand-crafted Japanese HPSG grammar that provides semantic information as well as linguistically motivated analysis of complex constructions. However, the evaluation of the grammar has not been done on domain-independent real-world texts such as newspaper articles. Although Bond et al. (2004) attempted to improve the coverage of the JACY grammar through the development of an HPSG treebank, they limited the target of their treebank annotation to short sentences from dictionary definitions. SLUNG (Mitsuishi et al., 1998) is an HPSG grammar whose coverage on real-world sentences is about 99%, but the grammar is *underspecified*, which means that the constraints of the grammar are not sufficient for conducting semantic analysis. By employing corpus-oriented development, we aim to develop a wide-coverage HPSG parser that enables semantic analysis of real-word texts.

## 3 Grammar Design

First, we provide a brief description of some characteristics of Japanese. Japanese is head final, and phrases are typically headed by function words. Arguments of verbs usually have no fixed order (this phenomenon is called *scrambling*) and are freely omitted. Arguments' semantic relations to verbs are chiefly determined by their head postpositions. For example, 'boku/*I* ga/*NOM* kare/*he* wo/*ACC* koroshi/*kill* ta/*DECL*' (I killed him) can be paraphrased as 'kare wo boku ga koroshi ta,' without changing the meaning.

The *case alternation* phenomenon must also be taken into account. Case alternation is caused by special auxiliaries "(sa)se" and "(ra)re," which are causative and passive auxiliaries, respectively, and the verbs change their subcategorization behavior when they are combined with these auxiliaries.

The following sections describe the design of our grammar. Especially, treatment of the scrambling and case alternation phenomena is provided in detail.

### 3.1 Fundamental Phrase Structures

Figure 2 presents the basic structure of signs of our grammar. The *HEAD* feature specifies phrasal categories, the *MOD* feature represents restrictions on the left and right modifiees, and the *VAL* feature encodes valence information. (For the explanation of the *BAR* feature, see the description of the *promo-*

Table 1: Schemata and their uses.

| schema name | common use of the rule |
|---|---|
| specifier-head | PP or NP + postposition<br>VP + verbal ending<br>NP + suffix |
| complement-head | argument (PP/NP) + verb |
| compound-noun | NP + NP |
| modifier-head | modifier + head |
| head-modifier | phrase + punctuation |
| promotion | promotes chunks to phrases |

*tion* schema below.) [1] For some types of phrases, additional features are specified as *HEAD* features.

Now, we provide a detailed explanation of the design of the schemata and how the features in Figure 2 work. The following descriptions are also summarized in Table 1.

**specifier-head schema** Words are first concatenated by this schema to construct basic word chunks. Postpositional phrases (PPs), which consist of postpositions and preceding phrases, are the most typical example of *specifier-head* structures. For postpositions, we specify a head feature *PFORM*, with the postposition's surface string as its value, in addition to the features in Figure 2, because differences of postpositions play a crucial role in disambiguating semantic-structures of Japanese. For example, the postposition 'wo' has a *PFORM* feature whose value is "wo," and it accepts an NP as its specifier. As a result, a PP such as "kare wo" inherits the value of *PFORM* feature "wo" from 'wo.'

The schema is also used when VPs are constructed from verbs and their endings (or, sometimes auxiliaries. See also Section 3.2).

**complement-head schema** This schema is used for combining VPs with their subcategorized arguments (see Section 3.2 for details).

**compound-noun schema** Because nouns can be freely concatenated to form compound nouns, a special schema is used for compound nouns.

**modifier-head schema** This schema is for modifiers and their heads. Binary structures that cannot be captured by the above three schemata are also considered to be modifier-head structures.[2]

**head-modifier schema** This schema is used when the *modifier-head* schema is not appropriate. In the current implementation, it is used for a phrase and its following punctuation.

**promotion schema** This unary schema changes the value of the *BAR* feature from *chunk* to *phrase*. The distinction between these two types of constituents is for prohibiting some kind of spurious ambiguities. For example, 'kinou/*yesterday* koroshi/*kill* ta/*DECL*' can be analyzed in two different ways, i.e. '(kinou (koroshi ta))' and '((kinou koroshi) ta).' The latter analysis is prevented by restricting "kinou"'s modifiee to be a *phrase*, and "ta"'s specifier to be a *chunk*, and by assuming "koroshi" to be a *chunk*.

### 3.2 Scrambling and Case Alternation

Scrambling causes problems in designing a Japanese HPSG grammar, because original HPSG, designed for English, specifies the subcategorization frame of a verb as an ordered list, and the semantic roles of arguments are determined by their order in the complement list.

Our implementation treats the complement feature as a list of semantic roles. Semantic roles for which verbs subcategorize are *agent*, *object*, and *goal*.[3] Correspondingly, we assume three subtypes of the *complement-head* schema: the *agent-head*, *object-head*, and *goal-head* schemata. When verbs take their arguments, arguments receive semantic roles which are permitted by the subcategorization of verbal signs. We do not restrict the order of application of the three types of *complement-head* schemata, so that a single verbal lexical entry can accept arguments that are scrambled in arbitrary order. In Figure 3, "kare ga" is a ga-marked PP, so it is analyzed as an agent of "koro(su)." [4]

Case alternation is caused by special auxiliaries "(sa)se" and "(ra)re." For instance, in 'boku/*I*

[1] The *CONTENT* feature, which should contain information about the semantic contents of syntactic entities, is ignored in the current implementation of the grammar.

[2] Current implementation of the grammar treats complex structures such as relative clause constructions and coordinations just the same as simple modification.

[3] These are the three roles most commonly found in EDR.

[4] We assume that a single semantic role cannot be occupied by more than one syntactic entities. This assumption is sometimes violated in EDR's annotation, causing failures in grammar extraction.

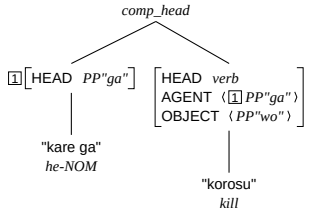


Figure 3: Verb and its argument.

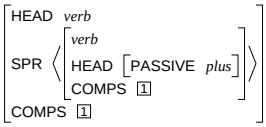


Figure 4: Lexical sign of "(ra)re".

ga/*NOM* kare/*he* ni/*DAT* korosa/*kill* re/*PASSIVE* ta/*DECL'* (I was killed by him), "korosa" takes a "ga"-marked PP as an object and a "ni"-marked PP as an agent, though without "(sa)re," it takes a "ga"-marked PP as an agent and a "wo"-marked PP as an object.

We consider auxiliaries as a special type of verbs which do not have their own subcategorization frames. They inherit the subcategorization frames of verbs.[5] To capture the case alternation phenomenon, each verb has distinct lexical entries for its passive and causative uses. This distinction is made by binary valued *HEAD* features, *PASSIVE* and *CAUSATIVE*. The passive (causative) auxiliary restricts the value of its specifier's *PASSIVE* (*CAUSATIVE*) feature to be *plus*, so that it can only be combined with properly case-alternated verbal lexical entries.

Figure 4 presents the lexical sign of the passive auxiliary "(ra)re." Our analysis of an example sentence is presented in Figure 5. Note that the passive auxiliary "re(ta)" requires the value of the *PASSIVE* feature of its specifier be *plus*, and hence "koro(sa)" cannot take the same lexical entry as in Figure 3.

## 4 Grammar Extraction from EDR

The EDR Japanese corpus consists of 207802 sentences, mainly from newspapers and magazines. The annotation of the corpus includes word segmentation, part-of-speech (POS) tags, phrase structure annotation, and semantic information.

[5]The control phenomena caused by auxiliaries are currently unsupported in our grammar.

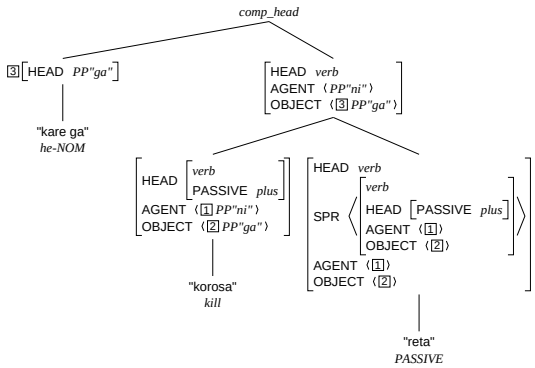


Figure 5: Example of passive construction.

The heuristic conversion of the EDR corpus into an HPSG treebank consists of the following steps. A sentence '((kare/NP-*he* wo/PP-*ACC*) (koro/VP-*kill* shi/VP-*ENDING* ta/VP-*DECL*))' ([I] killed him yesterday) is used to provide examples in some steps.

**Phrase type annotation** Phrase type labels such as NP and VP are assigned to non-terminal nodes. Because Japanese is head final, the label of the rightmost daughter of a phrase is usually percolated to its parent. After this step, the example sentence will be '((PP kare/NP wo/PP) (VP koro/VP shi/VP ta/VP)).'

**Assign head features** The types of head features of terminal nodes are determined, chiefly from their phrase types. Features specific to some categories, such as *PFORM*, are also assigned in this step.

**Binarization** Phrases for which EDR employs flat annotation are converted into binary structures. The binarized phrase structure of the example sentence will be '((kare wo) ((koro shi) ta)).'

**Assign schema names** Schema names are assigned according to the patterns of phrase structures. For instance, a phrase structure which consists of PP and VP is identified as a *complement-head* structure, if the VP's argument and the PP are coindexed. In the example sentence, 'kare wo' is annotated as 'koro"s object in EDR, so the *object-head* schema is applied to the root node of the derivation.

**Inverse schema application** The consistency of the derivation of the obtained HPSG treebank is ver-

ified by applying the schemata to each node of the derivation trees in the treebank.

**Lexicon Extraction** Lexical entries are extracted from the terminal nodes of the obtained treebank.

## 5 Disambiguation Model

We also train disambiguation models for the grammar using the obtained treebank. We employ log-linear models (Berger et al., 1996) for the disambiguation. The probability of a parse $P$ of a sentence $S$ is calculated as follows:

$$p(P|S) = \frac{\exp(\sum_i f_i(P)\lambda_i)}{\sum_{P'} \exp(\sum_i f_i(P')\lambda_i)}$$

where $f_i$ are feature functions, $\lambda_i$ are strengths of the feature functions, and $P'$ spans all possible parses of $S$. We employ Gaussian MAP estimation (Chen and Rosenfeld, 1999) as a criterion for optimizing $\lambda_i$. An algorithm proposed by Miyao et. al. (2002) provides an efficient solution to this optimization problem.

## 6 Experiments

Because the aim of our research is to construct a Japanese parser that can extract semantic information from real-world texts, we evaluated our parser in terms of its coverage and semantic-role identification accuracy. We also compare the accuracy of our parser with that of an existing statistical dependency analyzer, in order to investigate the necessity of further improvements to our disambiguation model.

The following experiments were conducted using the EDR Japanese corpus. An HPSG grammar was extracted from 51951[6] sentences of the corpus, and the same set of sentences were used as a training set for the disambiguation model. 47767 sentences (91.9%) of the training set were successfully converted into an HPSG treebank, from which we extracted lexical entries.

When we construct a lexicon from the extracted lexical entries, we reserved lexical entry templates for infrequent words as default templates for unknown words of each POS, in order to achieve sufficient coverage. The threshold for 'infrequent' words were determined to be 30 from the results of preliminary experiments.

We used 2079 EDR sentences as a test set. (Another set of 2078 sentences were used as a development set.) The test set is also converted into an HPSG treebank, and the conversion was successful for 1913 sentences. (We will call the obtained HPSG treebank the "test treebank.")

As features of the log-linear model, we extracted the POS of the head, template name of the head, surface string and its ending of the head, punctuation contained in the phrase, and distance between heads of daughters, from each sign in derivation trees. These features are used in combinations.

The coverage of the parser[7] on the test set was 95.3% (1982/2079). Though it is still below the coverage achieved by SLUNG (Mitsuishi et al., 1998), our grammar has richer information that enables semantic analysis, which is lacking in SLUNG.

We evaluated the parser in terms of its accuracy in identifying semantic roles of arguments of verbs. For each phrase which is in *complement-head* relation with some VP, a semantic role is assigned according to the type[8] of the *complement-head* structure. The performance of our parser on the test treebank was 63.8%/57.8% in precision/recall of semantic roles.

As most studies on syntactic parsing of Japanese have focused on *bunsetsu*-based dependency analysis, we also attempted an evaluation in this framework.[9] In order to evaluate our parser by *bunsetsu* dependency, we converted the phrase structures of EDR and the output of our parser into dependency structures of the right-most content word of each *bunsetsu*. *Bunsetsu* boundaries of the EDR sentences were determined by using simple heuristic rules. The dependency accuracies and the sentential accuracies of our parser and Kanayama et. al.'s analyzer are shown in Table 2. (*failure* sentences are not counted for calculating accuracies.) Our results were still significantly lower than those of

[6] We could not use the entire corpus for the experiments, because of the limitation of computational resources.

[7] Coverage of the parser can be somewhat lower than that of the grammar, because we employed a beam thresholding technique proposed by Tsuruoka et al. (Tsuruoka et al., 2004).

[8] As described in Section 3.2, there are three types of *complement-head* structures.

[9] *Bunsetsu* is a widely accepted syntactic unit of Japanese, which usually consists of a content word followed by a function word.

| | accuracy (dependency) | | accuracy (sentence) | | # failure |
|---|---|---|---|---|---|
| (Kanayama et al., 2000) | 88.6% | (23078/26062) | 46.9% | (1560/3326) | 1.4% (46/3372) |
| This paper | 85.0% | (13201/15524) | 37.4% | (705/1887) | 1.4% (26/1913) |

Table 2: Accuracy of dependency analysis.

Kanayama et. al., which are the best reported dependency accuracies on EDR.

This experiment revealed that the accuracy of our parser requires further improvement, although our grammar achieved high coverage. Our expectation is that incorporating grammar rules for complex structures which is ignored in the current implementation (e.g. control, relative clause, and coordination constructions) will improve the accuracy of the parser. In addition, we should investigate whether the semantic analysis our parser provides can contribute the performance of more application-oriented tasks such as information extraction.

## 7 Conclusion

We developed a Japanese HPSG grammar by means of the corpus-oriented method, and the grammar achieved the high coverage, which we consider to be nearly sufficient for real-world applications. However, the accuracy of the parser in terms of dependency analysis was significantly lower than that of the existing parser. We expect that the accuracy can be improved through further elaboration of the grammar design and disambiguation method.

## References


Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1).

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 *Proc. of IJCNLP-04*.

J. Bresnan and R. M. Kaplan. 1982. Introduction: Grammars as mental representations of language. In *The Mental Representation of Grammatical Relations*. MIT Press.

S. Chen and R. Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. In *Technical Report CMUCS*.

Julia Hockenmaier and Mark Steedman. 2002. Acquiring Compact Lexicalized Grammars from a Cleaner Treebank. In *Proc. of Third LREC*.

Hiroshi Kanayama, Kentaro Torisawa, Mitsuishi Yutaka, and Jun'ichi Tsujii. 2000. A Hybrid Japanese Parser with Hand-crafted Grammar and Statistics. In *Proc. of the 18th COLING*, volume 1.

Yutaka Mitsuishi, Kentaro Torisawa, and Jun'ichi Tsujii. 1998. HPSG-Style Underspecified Japanese Grammar with Wide Coverage. In *Proc. of the 17th COLING–ACL*.

Yusuke Miyao and Jun'ichi Tsujii. 2002. Maximum Entropy Estimation for Feature Forests. In *Proc. of HLT 2002*.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. 2004. Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In *Proc. of IJCNLP-04*.

National Institute of Information and Communications Technology. 2002. EDR Electronic Dictionary Version 2.0 Technical Guide.

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

Y. Schabes, A. Abeille, and A. K. Joshi. 1988. Parsing Strategies with 'Lexicalized' Grammars: Application to Tree Adjoining Grammars. In *Proc. of the 12th COLING*.

Melanie Siegel and Emily M. Bender. 2002. Efficient Deep Processing of Japanese. In *Proc. of the 3rd Workshop on Asian Language Resources and International Standardization. COLING 2002 Post-Conference Workshop, August 31*.

Mark Steedman. 2001. *The Syntactic Process*. MIT Press.

Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2004. Towards efficient probabilistic HPSG parsing: integrating semantic and syntactic preference to guide the parsing. In *Proc. of IJCNLP-04 Workshop: Beyond shallow analyses - Formalisms and statistical modeling for deep analyses*.