Res on Lang and Comput (2010) 8:1–22
DOI 10.1007/s11168-010-9069-7

Exploiting Semantic Information for HPSG Parse Selection

Sanae Fujita · Francis Bond · Stephan Oepen · Takaaki Tanaka

Received: 28 September 2009 / Accepted: 16 September 2010 / Published online: 20 October 2010
© Springer Science+Business Media B.V. 2010

Abstract In this article, we investigate the use of semantic information in parse selection. We show that fully disambiguated sense-based semantic features smoothed using ontological information are effective for parse selection. Training and testing was undertaken using definition and example sentences taken from a Japanese dictionary corpus (Hinoki), which is manually annotated with senses. A model employing both syntactic and semantic information provides better parse selection accuracy than a model using only syntactic features.

Keywords HPSG · Parse selection · Semantic information

This article presents an updated and extended version of results first published by Fujita et al. (2007).

S. Fujita (✉)
NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation,
Kyoto, Japan
e-mail: sanae@cslab.kecl.ntt.co.jp

F. Bond
Division of Linguistics and Multilingual Studies, Nanyang Technological University,
Singapore, Singapore
e-mail: bond@ieee.org

S. Oepen
Department of Informatics, University of Oslo, Oslo, Norway
e-mail: oe@ifi.uio.no

T. Tanaka
NTT West, Osaka, Japan
e-mail: taka.tanaka@rdc.west.ntt.co.jp

1 Introduction

In this paper we investigate the use of semantic information in parse selection, i.e. probabilistic disambiguation of competing grammatical analyses. Recently, significant improvements have been made in combining symbolic and statistical approaches to various natural language processing tasks. In parsing, for example, symbolic grammars are combined with stochastic models (Riezler et al. 2002; Oepen et al. 2004; Malouf and van Noord 2004; Miyao and Tsujii 2008). Much of the benefit gained with statistical parsing using lexicalized models comes from the use of a small set of function words (Klein and Manning 2003). Features based on content words provide little improvement, presumably because the data is too sparse. For example, in the Penn Treebank which is widely used to train and test statistical parsers, *stocks* and *skyrocket*[1] never appear together. However, the superordinate concepts *capital* (⊃ *stocks*) and *move upward* (⊃ *sky rocket*) frequently appear together, which suggests that employing word senses and their hypernyms as features may be useful.

Another example is given in (1):[2]

(1)	私は	大阪と	京都に	行き、	彼は	社長と	京都に
	watashi-wa	oosaka-to	kyouto-ni	iki,	kare-wa	shachou-to	kyouto-ni
	I-TOP	Osaka-?	Kyoto-LOC	go,	he-TOP	president-?	Kyoto-LOC
	行った。						
	itta.						
	went.						

‘I went to Osaka and Kyoto and he went to Kyoto with the president.’

Both clauses share the surface structure: A-TOP B-*to* C-LOC Verb. The particle と *to* is ambiguous between a conjunction (where B and C coordinate) and a postposition, where B attaches to the verb phrase. 大阪 *oosaka* “Osaka” and 社長 *shachou* “president” are both nouns, with similar syntactic behavior. Semantic information disambiguates: if B is a place then it is more likely to form a conjunction with another place (*and*), whereas if B is a person then it is more likely to be a comitative modifier (*with*).

However, to date, there have been few reports on the combination of sense information with symbolic grammars and statistical models. We hypothesize that one reason for the lack of success is that there are few resources annotated with both syntactic and semantic information. In this paper, we use a treebank with both syntactic information (parses from a Head-Driven Phrase Structure Grammar; HPSG: Pollard and Sag 1994) and semantic information (manually annotated with lexical senses: Bond et al. 2006). We use this to train parse selection models using both syntactic and semantic features. A model trained using syntactic features combined with fully disambiguated semantic information outperforms a model using purely syntactic information by a wide margin; 69.4% sentence parse accuracy versus 63.8%, tested on definition sentences.

[1] In this case, *skyrocket* is a verb, with the meaning “shoot up abruptly, like a rocket”.

[2] We use the following abbreviations: TOP: topic; NOM: nominative; ACC: accusative; LOC: locative.

A parse is judged correct if and only if the first ranked parse for the entire sentence is identical to the gold standard parse.

In the next section, we introduce the resources which we use. Then, in Section 3, we describe the parse selection model and feature types. In Section 4, we describe the evaluation method. Then in Section 5, we present and discuss the experimental results. In Section 6, we compare our results to related work.

2 The Hinoki Corpus

The earliest corpus with both syntactic and semantic annotation we know of is the subset of the Brown corpus annotated with both trees (Penn Treebank) and WordNet senses (SemCor: Fellbaum (1998)). However, the subset is quite small (8,700 sentences). There is currently a larger project under way for English, the OntoNotes project (Hovy et al. 2006) which is combining sense tags with a larger section of the Penn Treebank. In this paper we use Japanese data from the Hinoki corpus consisting of around 95,000 dictionary definition and example sentences (Bond et al. 2006) which is annotated with both syntactic parses and senses from the same dictionary.

2.1 Syntactic Annotation

The syntactic annotation in Hinoki is *grammar-based corpus annotation* accomplished by selecting the best parse (or parses) from the full analyses derived by using a broad-coverage symbolic grammar. The grammar is an HPSG implementation (JACY: Siegel and Bender 2002), which provides a high level of detail, marking not only dependency and constituent structure but also detailed semantic relations. As the grammar is based on a monostratal theory of grammar (HPSG: Pollard and Sag 1994), annotation by manual disambiguation determines the syntactic and semantic structure simultaneously. Using a grammar helps treebank consistency—all sentences annotated are guaranteed to have well-formed parses. The flip side to this is that any sentences that the parser cannot parse remain unannotated, at least unless we were to employ a full manual mark-up of their analyses. The actual annotation process uses the same tools as the Redwoods treebank of English (Oepen et al. 2004).

A (simplified) example of a parse is given in Fig. 1. The text is taken from a dictionary definition, and is a noun phrase rather than a sentence. Using JACY, there were four parses for the definition sentence. The correct parse in this context, shown as a phrase structure tree, is the one in Fig. 1. There are two sources of ambiguity: a conjunction and a relative clause. The parser allows the conjunction to combine 電車 *densha* "train" with 人 *hito* "person" (high attachment) as well as the preferred low attachment to 自動車 *jidōsha* "car". In Japanese, relative clauses can have gapped and non-gapped readings. In the gapped reading (selected here), 人 *hito* "person"is the subject of 運転 *unten* "drive". In the non-gapped reading there is some underspecified relation between the head noun and the verb phrase. This is similar to the difference in the two readings of *the day he knew* in English: "the day that he knew about" (gapped) vs "the day on which he knew (something)" (non-gapped). The four parses can be thus be glossed as: "a person who drives trains and cars" (correct), "# trains and a person

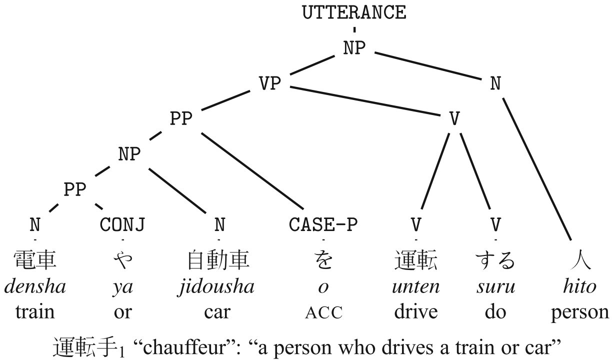

Fig. 1 Syntactic view of the definition of 運転手1 *untensha* “chauffeur”

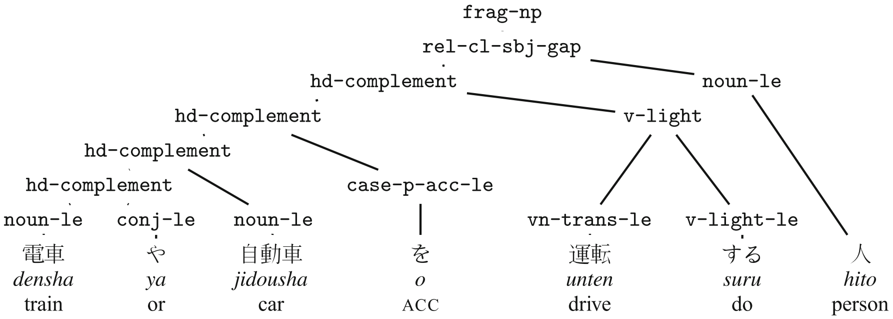

Fig. 2 Derivation tree of the definition of 運転手1 *untenshu* “chauffeur”

who drives”, “# a person that someone drives trains and cars”, “# trains and a person that someone drives cars”. Such semantic ambiguity is resolved by selecting the correct derivation tree, the primary (internal) representation produced by the grammar, that includes the applied rules when building the tree (Fig. 2). In the derivation tree phrasal nodes are labeled with identifiers of grammar rules, and (pre-terminal) lexical nodes with class names for types of lexical entries.

In addition to the syntactic tree, the parser also gives a rich semantic representation—primarily capturing grammaticalized propositional structure—as shown in Fig. 3. Our specific meaning representation language is Minimal Recursion Semantics (MRS: Copestake et al. 2005). We simplify this into a (semantic) dependency representation (Oepen and Lønning 2006), further abstracting away from scopal relations and quantification, as shown in Fig. 4. One of the advantages of the HPSG sign is that it contains all this information, making it possible to extract the particular view needed. In order to link to other resources, such as the sense annotation, elementary predicates are labeled with pointers back to their position in the original surface string (i.e. a character range denoting a sub-string, a mechanism dubbed *characterization*). For example, the predicate `densha_n_1` links to the surface characters between positions 0 and 3: 電車. The semantic view shows that some ambiguity has been resolved

```
TEXT    電車や自動車を運転する人
TOP     h1
RELS    {
         [proposition_m_rel    [unknown_rel    [_densha_n    [udef_rel     [_ya_p           [udef_rel
          LBL   h1              LBL   h4        LBL   h6      LBL  h8       LBL      h11      LBL  h15
          ARG0  e2              ARG0  e2        ARG0  x7]     ARG0 x7       ARG0     x13      ARG0 x12
          MARG  h3]             ARG   x5]                     RSTR h9       L-HNDL   u13      RSTR h16
                                                              BODY h10]     L-INDEX  x7       BODY h17]
                                                                            R-HNDL   u14
                                                                            R-INDEX  x12]

         [_jidousha_n  [udef_rel   [_unten_s                     [_hito_n    [udef_rel   [proposition_m_rel
          LBL   h18     LBL  h19    LBL   h22                     LBL  h24    LBL  h25    LBL   h10001
          ARG0  x12]    ARG0 x12    ARG0  e23 tense=present       ARG0 x5]    ARG0 x5     ARG0  e23 tense=present
                        RSTR h20    ARG1  x5                                  RSTR h26    MARG  h28]
                        BODY h21]   ARG2  x13]                                BODY h27]
        }
HCONS   {h3 qeq h4,h9 qeq h6,h16 qeq h11,h20 qeq h18,h26 qeq h24,h28 qeq h22}
ING     {h24 ing h10001}
```

Fig. 3 Semantic view of the definition of 運転手$_1$ *untenshu* “chauffeur”

```
1   e2:unknown<0:13>[ARG x5:hito_n_1]
2   x7:densha_n_1<0:3>[]
3   x12:jidousha_n_1<4:7>[]
4   x13:ya_p_conj<0:4>[LIDX x7:densha_n_1, RIDX x12:jidousha_n_1]
5   e23:unten_s_2<8:10>[ARG1 x5:hito_n_1]
6   e23:unten_s_2<8:10>[ARG2 x13:ya_p_conj]
```

We assign a number per line to facilitate citation.

Fig. 4 Elementary dependency view (EDs) of the definition of 運転手$_1$ *untenshu* “chauffeur”

```
INDEX     運転手  untenshu
POS       noun
SENSE 1   [DEFINITION      [電車1 や 自動車1 を 運転1 する 人4 。
                            a person who drives trains and cars.]
           EXAMPLE         [大きく5 なったら電車1 の運転手1 に成る6 のが夢3 です。
                            I dream of growing up and becoming a train driver.]
           HYPERNYM        人4  hito “person”
           SEMANTIC CLASS  ⟨292:driver⟩ (⊂ ⟨4:people⟩)
           WORDNET         motorman1]
```

Fig. 5 Dictionary entry for 運転手$_1$ *untenshu* “chauffeur”

that is not visible in the purely syntactic view, namely that the relative clause is gapped (and thus the semantic variable of the conjoined NP is the ARG1 of the relation introduced by the verb).

2.2 Sense Annotation

The lexical semantic annotation uses the sense inventory from Lexeed (Kasahara et al. 2004), which in turn is based on the most familiar 30,000 words from a medium-sized desktop dictionary (Kindaichi and Ikeda 1988). All the words in this fundamental vocabulary are manually annotated with their sense (Bond et al. 2006). For example, the word 大きい *ookii* “big” is annotated as sense 5 in the example sentence in Fig. 5, with the meaning “older” (its surface form is 大きく *ookiku*).

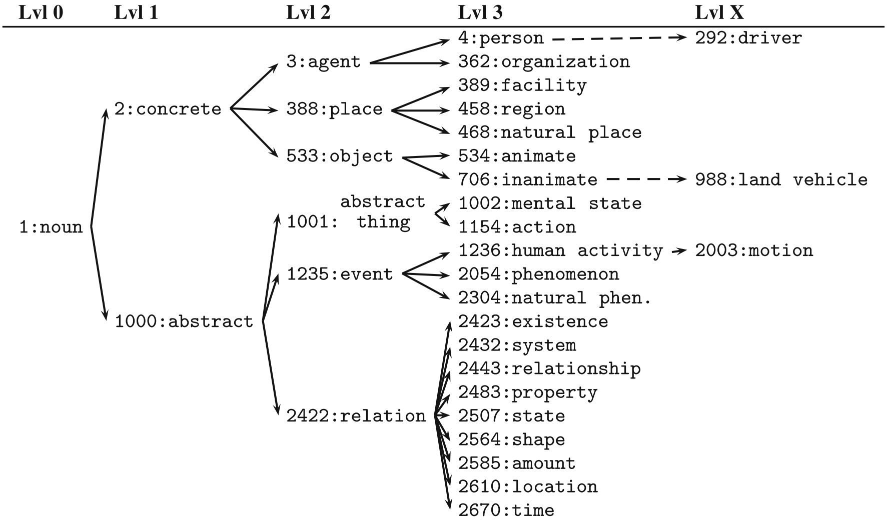

Fig. 6 Top 4 levels of the GoiTaikei Ontology. *Note* The *solid arrows* show the direct parent and child relationship. *Dashed arrows* show indirect relationships (skipping some levels)

Figure 5 shows a complete entry from Lexeed. Each entry contains the word itself, its part of speech, and its lexical type(s) in the grammar. Each sense then contains definition and example sentences, links to other senses in the lexicon (such as hypernym), and links to other resources, such as the Goi-Taikei Japanese Lexicon (Ikehara et al. 1997) and WordNet (Fellbaum 1998). Each content word of the definition and example sentences is annotated with sense tags from the same lexicon.

The word senses are further linked to semantic classes in a Japanese ontology. The ontology, Goi-Taikei, consists of a hierarchy of 2,710 semantic classes, defined for over 264,312 nouns, with a maximum depth of 12 (Ikehara et al. 1997) (Level 0 to 11). We show the top four levels and some lower classes of the Goi-Taikei common noun ontology in Fig. 6. The semantic classes are principally defined for nouns (including verbal nouns), although there is some information about verbs and adjectives.

3 Parse Selection

In the past decade, research groups working within grammatical frameworks like Lexical-Functional Grammar (LFG: Riezler et al. 2002), and HPSG (Malouf and van Noord 2004; Oepen et al. 2004) have successfully integrated broad-coverage (manually developed), symbolic grammars with sophisticated statistical parse selection models. In this setup, the linguistic grammar constrains the space of possible analyses, while the models provide a probability distribution over competing hypotheses. Parse selection approaches for these frameworks predominantly use discriminative Maximum Entropy (ME) models, where the probability of each parse tree, given an input string, is estimated on the basis of selected properties (called features) of the tree (Abney 1997; Johnson et al. 1999). Such features, in principle, are not restricted in their domain of locality, and enable the parse selection process to take into account properties that extend beyond local contexts (i.e. sub-trees of depth one).

Similar in spirit to much of this work, and combining the broad-coverage JACY grammar and the Hinoki corpus, we construct a parse selection model on top of the symbolic grammar. Given a set of candidate analyses (for some Japanese string) according to JACY, the goal is to rank parse trees by their probability: training a stochastic parse selection model on the available treebank, we estimate the statistics of various features of the candidate analyses from the treebank. Therefore, the definition and selection of features is a central parameter in the design of an effective parse selection model.

3.1 Syntactic Features

The first model that we trained uses syntactic features defined over HPSG derivation trees as summarized in Table 1. For the closely related purpose of parse selection over the English Redwoods treebank, Toutanova et al. (2005) train a discriminative log-linear model, using features defined over *derivation trees* with non-terminals representing the *construction types* and *lexical types* of the HPSG grammar. The basic feature set of our parse selection model for Japanese is defined in the same way (corresponding to the PCFG-S model reported by Toutanova et al. 2005): each feature captures a sub-tree from the derivation limited to depth one.

Table 1 shows a sub-set of example features extracted from our running example (Fig. 2 above) in our ME models, where the feature template #1 corresponds to local derivation sub-trees. The first column shows the identifier of the feature template corresponding to each example; in the examples, the first integer value is a parameter of the feature templates, i.e. the depth of grandparenting (types #1 and #2: see 3.1.1) or *n*-gram size (types #3 and #4). The special symbols △ and ◁ denote the root of the tree and left periphery of the yield, respectively. We will refer to the parse selection model that uses only local structural features as SYN-1.

Table 1 Example structural features (SYN-1, SYN-GP and SYN-ALL) extracted from the derivation tree in Fig. 2

#	Sample Features
1	⟨0 rel-cl-sbj-gap hd-complement noun-le⟩
1	⟨1 frag-np rel-cl-sbj-gap hd-complement noun-le⟩
1	⟨2 △ frag-np rel-cl-sbj-gap hd-complement noun-le⟩
2	⟨0 rel-cl-sbj-gap hd-complement⟩
2	⟨0 rel-cl-sbj-gap noun-le⟩
2	⟨1 frag-np rel-cl-sbj-gap hd-complement⟩
2	⟨1 frag-np rel-cl-sbj-gap noun-le⟩
3	⟨1 conj-le ya⟩
3	⟨2 noun-le conj-le ya⟩
3	⟨3 ◁ noun-le conj-le ya⟩
4	⟨1 conj-le⟩
4	⟨2 noun-le conj-le⟩
4	⟨3 ◁ noun-le conj-le⟩

3.1.1 Dominance Features

To reduce the effects of data sparseness, feature type #2 in Table 1 provides a back-off to derivation sub-trees, where the sequence of daughters is reduced to just the head daughter in each local context. Conversely, to facilitate the sampling of larger contexts than simply sub-trees of depth one, feature template #1 allows optional grandparenting. In addition to a parent and its children (`rel-cl-sbj-gap hd-complement noun-le`), we include an upwards chain of dominating nodes or grandparents (△ `frag-np`) in some features. In our experiments, we found that grandparenting of up to three dominating nodes gave the best balance between enlarged context and data sparseness. Henceforth, we will refer to our basic model SYN-1 enriched with these features as SYN-GP.

3.1.2 N-Gram Features

In addition to these dominance-oriented features taken from the derivation trees of each parse tree, our models also include more surface-oriented features, viz. n-grams of lexical types with or without lexicalization. Feature type #3 in Table 1 defines n-grams of variable size, where (in a loose analogy to part-of-speech tagging) sequences of lexical types capture syntactic category assignments, similarly to the use of supertagging to constrain parsing (Ciaramita and Altun 2006). Feature templates #3 and #4 only differ with regard to lexicalization, as the former includes the surface token associated with the rightmost element of each n-gram (loosely corresponding to the emission probabilities in an HMM tagger). We used a maximum n-gram size of two in the experiments reported here, again due to its empirically determined best overall performance. Henceforth, we will refer to the model SYN-GP enriched with these features as SYN-ALL.

3.1.3 Constituent Weight

We also investigated features based on constituent weight (the ratio of the lengths of the children of binary rules). However, we have been unable to derive an improvement in parse selection performance from the addition of these additional features on top of the SYN-ALL configuration (and only a comparatively moderate improvement on top of the basic model SYN-1). We therefore did not use these features in the final configuration.

3.2 Semantic Features

Using our syntactic feature configurations as a baseline, from here, we define semantic features for investigating the effectiveness of semantic information in parse selection. Recall that, to define the semantic parse selection features, we use the reduction of the full semantic representation (MRS) into ‘variable-free’ *elementary dependencies*, as proposed by Oepen and Lønning (2006). The conversion centers on a notion of one *distinguished variable* in each semantic relation. For most types of relations, the

Table 2 Example semantic features (SEM-Dep) extracted from the MRS in Fig. 4

#	Sample Features
20	⟨0 unten_s_2 ARG1 hito_n_1 ARG2 ya_p_conj⟩
20	⟨0 ya_p_conj LIDX densha_n_1 RIDX jidousha_n_1⟩
21	⟨1 unten_s_2 ARG1 hito_n_1⟩
21	⟨1 unten_s_2 ARG2 ya_p_conj⟩
21	⟨1 ya_p_conj LIDX densha_n_1⟩
21	⟨1 ya_p_conj RIDX jidousha_n_1⟩
22	⟨2 unten_s_2 hito_n_1 ya_p_conj⟩
23	⟨3 unten_s_2 hito_n_1⟩
23	⟨3 unten_s_2 jidousha_n_1⟩
24	⟨1 unten_s_2 ARG2 densha_n_1⟩
24	⟨1 unten_s_2 ARG2 jidousha_n_1⟩
…	…

distinguished variable corresponds to the main index (ARG0 in the examples above), e.g. an event variable for verbal relations and a referential index for nominals. Setting aside quantifiers, a variable will be the main index for one and only one relation.[3] Therefore, an MRS can be broken down into a set of basic dependency tuples of the form shown in Fig. 4. To extract such tuples from an MRS—based on the distinguished variable notion—each variable is coupled with its 'representative' relation. In a few corner cases where the above uniqueness constraint on the introduction of main indices is not maintained, there usually exist linguistically motivated disambiguation heuristics that allow us to select a unique distinguished variable—for example preferring semantic heads over (intersective) modifiers in a cluster of logically conjoined relations. In a nutshell, elementary dependency tuples localize semantic argumenthood at the level of semantic predicates.

All predicates are indexed to the position of the word or words that introduced them in the input sentence ($\langle i : j \rangle$). This allows us to link them to the sense annotations in the corpus.

3.2.1 Basic Semantic Dependency Features

The basic semantic model, SEM-Dep, consists of features based on a predicate and its arguments taken from the elementary dependencies. For example, consider the dependencies given in Fig. 4. The predicate *unten_s_2* has two arguments: ARG1 *hito_n_1* "person" (line 5 in Fig. 4) and ARG2 *ya_p_conj* "conjunction"(line 6), which leads to *densha_n_1* and *jidousha_n_1* (line 4).

From these, we produce several features (see Table 2). One includes all the arguments and their labels (feature template #20). We also produce various back-offs:

[3] We assume that adjectives and adverbs have distinguished event-like variables of their own, which can be independently motivated by their predicative use.

template #21 introduces only one argument at a time, template #22 drops argument labels, template #23 provides one unlabeled argument at a time. Furthermore, we expand out coordinate conjunctions, replacing them with their conjuncts (line 4 in Fig. 4) as template #24).

Each combination of a predicate and its related argument(s) becomes a feature. These resemble the basic semantic features used by Toutanova et al. (2005), with the addition of the expanded conjunctions (such as #24).

We further simplify these features by collapsing some non-informative predicates, e.g. the `unknown` predicate used in fragments and `message` predicates, which encode illocutionary force (Ginzburg and Sag 2000).

3.2.2 Word Sense Based Features

To investigate the effectiveness of lexical semantic information, we create features based on word senses and semantic classes. Here, the general idea is that in the word sense model (SEM-WS), these features provide more specific information. These features should allow the learner to learn the differences between word senses not just words.

We do semantic smoothing with the semantic classes (SEM-Class), where senses are binned into only 2,710 classes. We further smooth these features by replacing the semantic classes with their hypernyms (SEM-L, SEM-VC). There are two major advantages to the underspecified semantic classes: First, they allow generalization and therefore reduce data sparseness. Second, we expect that they will be more robust to errors in word sense disambiguation, because fine grained sense distinctions can be ignored. This will be helpful in future work when we use automatically disambiguated senses instead of the gold standard annotations.

Mapping Predicates to Word Sense / Semantic Class / Superordinate Semantic Class The Hinoki corpus is annotated with word senses. These are informative, but very sparse. We therefore converted the word senses into semantic classes via the links defined in the Lexeed Dictionary. Then we backed-off by replacing the actual semantic classes with superordinate semantic classes via the hierarchy of Goi-Taikei.

For example, for the definition sentence of 運転手$_1$ *untenshu* "chauffeur" (Fig. 5), the word sense for predicate `densha_n_1` 電車$_1$ *densha* "train", and the semantic class for 電車$_1$ *densha* "train" is ⟨**`988:land vehicle`**⟩. Further, at level 3 of Goi-Taikei (see Fig. 6), the superordinate class for ⟨**`988:land vehicle`**⟩ is ⟨**`706:inanimate`**⟩. So we replace predicates with word senses, then semantic classes, and then superordinate semantic classes as shown in Fig. 7, which shows the predicate, word senses, and actual semantic class for each content word and the superordinate semantic classes at levels 2 to 5. The more general sense is shown lower here, thus ⟨**`988:land vehicle`**⟩ ⊂ ⟨**`986:vehicle`**⟩ ⊂ ⟨**`760:artifact`**⟩ ⊂ ⟨**`706:inanimate`**⟩ ⊂ ⟨**`533:object`**⟩. Predicates are binned into 9 distinct classes at level 2, 30 classes at level 3, 136 classes at level 4, and 392 classes at level 5. The sample features of SEM-Class are shown in Table 3. We use superordinate semantic classes to make features in the same way as in Table 3.

predicate	densha_n_1	ya_p_conj	jidousha_n_1	-
word sense	電車$_1$	や	自動車$_1$	を
gloss	trains	or	cars	ACC
sem. class	⟨988:land vehicle⟩	-	⟨988:land vehicle⟩	-
Lvl 5	⟨986:vehicle⟩	-	⟨986:vehicle⟩	-
Lvl 4	⟨760:artifact⟩	-	⟨760:artifact⟩	-
Lvl 3	⟨706:inanimate⟩	-	⟨706:inanimate⟩	-
Lvl 2	⟨533:object⟩	-	⟨533:object⟩	-

predicate	unten_s_2		hito_n_1
word sense	運転$_1$	する	人$_4$
gloss	drive	do	person
sem. class	⟨2003:motion⟩	-	⟨4:person⟩
Lvl 5	⟨1920:labor⟩	-	⟨4:person⟩
Lvl 4	⟨1560:act/conduct⟩	-	⟨4:person⟩
Lvl 3	⟨1236:human activity⟩	-	⟨4:person⟩
Lvl 2	⟨1235:event⟩	-	⟨3:agent⟩

Fig. 7 Mapping predicate to word sense, etc.: Definition of 運転手$_1$ *untenshu* “chauffeur”

Table 3 Example Semantic Features (SEM-Class) using Goi-Taikei Semantic Classes

#	Sample Features
40	⟨0 unten_s_2 ARG1 ⟨**4:people**⟩ ARG2 ⟨**988:land vehicle**⟩⟩
40	⟨1 ⟨**2003:steering, control, …**⟩ ARG1 ⟨**4:people**⟩ ARG2 ⟨**988:land vehicle**⟩⟩
40	⟨1 ⟨**2003:steering, control, …**⟩ ARG1 ⟨**4:people**⟩ ARG2 ⟨**988:land vehicle**⟩⟩
40	⟨0 ya_p_conj LIDX ⟨**988:land vehicle**⟩ RIDX ⟨**988:land vehicle**⟩⟩
41	⟨2 unten_s_2 ARG1 ⟨**4:people**⟩⟩
41	⟨2 unten_s_2 ARG2 ⟨**988:land vehicle**⟩⟩
…	…

Of course, although several polysemous senses belong to the same semantic class, the majority of these senses are linked into different semantic classes. We show the number of lexeed word senses per semantic class in Table 4. For example, of all the polysemous senses (48,180), 56.7% of the word senses will be completely disambiguated by the superordinate semantic class at level 5. In addition, even if they can’t be completely disambiguated, we expect that fine grained sense distinctions will always be essential for parse selection.

Frequency-based Selection of Superordinate Semantic Classes Selection There is no guarantee that adopting a set depth gives us the best distribution of semantic classes. In this section we investigate another method of selecting superordinate semantic classes (SEM-VC). With SEM-L, we use the semantic classes at a given level, but for SEM-VC, we use all the classes that occur above a certain frequency threshold. Semantic classes have a very unbalanced distribution. We use frequent semantic classes as they are, but merge infrequent semantic classes into superordinate classes until the summed frequency is above the threshold, as shown in Fig. 8. This threshold was determined empirically.

Table 4 No. of Word Senses of Lexeed per Semantic Class (at each Level)

No. of Word Senses per Class	Semantic Classes									
	ALL		Lvl 5 (392)		Lvl 4 (136)		Lvl 3 (30)		Lvl 2 (9)	
	No.	%	No.	%	No.	%	No.	%	No.	%
1	32,167	66.8	27,316	56.7	20,775	43.1	16,944	35.2	10,582	22.0
2	11,606	24.1	14,078	29.2	15,852	32.9	17,106	35.5	18,236	37.8
3	2,769	5.7	3,897	8.1	5,244	10.9	6,084	12.6	7,344	15.2
4	900	1.9	1,264	2.6	2,080	4.3	2,628	5.5	3,680	7.6
≥ 5	738	1.5	1,625	3.4	4,229	8.8	5,418	11.2	8,338	17.3
Total	48,180	100	48,180	100	48,180	100	48,180	100	48,180	100

```
Step1: For each semantic class ($C$), Count $freq(C)$ in the training corpus.

Step2: For each Level ($Lvl$), from deeper to higher
For each semantic class ($C$) at the $Lvl$
  – if $freq(C) > threshold$
      – Use this class $C$
  – else
      – merge into its parent class ($P$) (add $freq(C)$ into $freq(P)$) (do NOT use this class)
```

Fig. 8 Merging into superordinate semantic classes above a certain frequency

For example, if the threshold is 3,000,[4] for ⟨**2003:steering, control, operation**⟩ (at Level 9), $freq(C2003^{5})$ is 96 (<3,000), so it is merged into its parent ⟨**2002:work/ manipulation**⟩ (at Level 8). But $freq(C2002)$ remains 350 (<3,000), so it is merged into its parent ⟨**2001:operation/handling**⟩ (at Level 7). Then $freq(C2001)$ is 3,327 ($>=$ 3,000), so we use ⟨**2001:operation/handling**⟩ insted of ⟨**2003:steering, control, operation**⟩, ⟨**2002:work/manipulation**⟩, and so on. In the same way, ⟨**988:land vehicle**⟩ (Level 7) is merged into ⟨**760:artifact**⟩ (Level 4).

3.2.3 Valency Dictionary Compatibility

Another source of knowledge that we use is information about stereotypical verb usage (Semantic Patterns). This is encoded as valency information (SP), taken from the Japanese side of the Goi-Taikei Japanese-English valency dictionary as extended by Fujita and Bond (2007). This valency dictionary contains detailed information about the argument properties of verbs and adjectives, including subcategorization and selectional preferences. Simplified entries for two senses of the Japanese side of 運転する *unten-suru* “drive”are shown in Figs. 9 and 10, each with a unique Pattern ID (PID). Here the dictionary entry (Fig. 9) tells us that the subject of one sense of

[4] Using the **example** corpus.

[5] C2003 is ⟨**2003:steering,control,operation**⟩, C2002 is ⟨**2002:work/manipulation**⟩, and C2001 is ⟨**2001:operation/handling**⟩.

PID: 300513

┌ **N1** ⟨`4:people`⟩ が*ga* NOM
├ **N2** ⟨`986:vehicle`⟩ を*o* ACC
└ 運転する*unten-suru* “drive”

Fig. 9 Entry in valency dictionary for 運転する *unten-suru* “N1 drive N2” (1)

PID: 300514

┌ **N1** ⟨`3:agents, 962:machinery`⟩ が*ga* NOM
├ **N2** ⟨`962:machinery`⟩ を*o* ACC
└ 運転する*unten-suru* “run”

Fig. 10 Entry in valency dictionary for 運転する *unten-suru* “N1 drive N2” (2)

How to Select the Best Pattern:

Step1: For each predicate (*Pred*), (e.g. 運転する*unten_s_2* “drive” in Figure 4)
Extract candidate valency dicrionary entries (*Pat*s)
whose lemma is the same as the predicate. (e.g. PID: 300513 and 300514 in Figure 9, 10)

- if the *Pred* is intransitive
  - filter out transitive *Pat*s (including accusative)
- else if the *Pred* is transitive
  - filter out intransitive *Pat*s (not including accusative)

Step2: For each candidate Pat_i, Calculate $score(Pat)$ (e.g. for 300513 and 300514)
For each argument (N_j) of Pat_i (e.g. for N1 and N2)

- if *Pred* has a argument whose case-markers is the same (e.g. ARG1 for N1, ARG2 for N2)
  - Add $score(N_j)$ to $score(Pat_i)$ (see below)
- else if *Pred* has a comitative phrase whose case-markers is the same
  - Add $score(N_j)$ to $score(Pat_i)$
- else
  - Add 0 to $score(Pat_i)$

Step3: Select the *Pat* with the highest $score(Pat)$ (e.g. select 300514 ($score(300514) > score(300513)$))

Calculation Method of $score(N_j)$:

- if N_j’s semantic restriction is a literal word
  - if an input word exactly matches it
    $score(N_j) = 10000$ (to chose this *Pat*)
- else (e.g. ⟨`986:vehicle`⟩ for N2 of 300514)
  - Give a value according to the level of the matching semantic restriction
    (from 100 at a leaf level to 60 for the top level)
    (e.g. ARG2’s RIDX:*jidousha_n_1* “car” ⊂ ⟨`988:land vehicle`⟩, which is subsumed by ⟨`986:vehicle`⟩; and matched on leaf node: the score is 100)
  - Then, weight the score according to the matched case-role:

	Role	Weight	Example
N1	agent	×1.5	
N2, N3	patient/theme	×2	100 ×2 for N2 of 300514
N4, N5	source, goal	×1	
N6–N8	manner/instrument	×0.8	
other		×0.5	

Fig. 11 Finding the most appropriate pattern and scoring its compatibility

運転する *unten-suru* “drive” is normally a person ⟨**`4:people`**⟩ and the thing driven is normally a vehicle ⟨**`986:vehicle`**⟩.

Each entry has a predicate and several case-slots. Each case-slot has information such as grammatical function, case-marker, case-role (N1, N2, ...), and semantic restrictions. The semantic restrictions are defined by either Goi-Taikei’s semantic classes or actual words (e.g. for idioms). On the Japanese side of Goi-Taikei’s valency

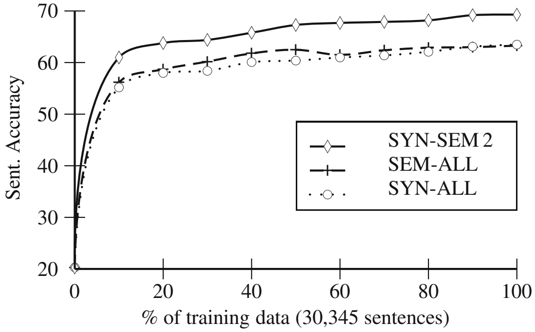

Fig. 12 Learning curves (Definitions)

Table 5 Example semantic features (SP) of Valency Dictionary Compatibility

#	Sample Features
31	⟨0 High⟩
31	⟨1 PID:300513 High⟩
31	⟨2 2⟩
31	⟨3 RIDX:High⟩
31	⟨4 PID:300513 RIDX:High⟩
32	⟨1 unten_s_2 High⟩
32	⟨4 unten_s_2 RIDX:High⟩
33	⟨5 N1 Child High⟩
33	⟨7 Child⟩
…	

dictionary, there are 10,146 types of verbs giving 18,512 entries and 1,723 types of adjectives giving 2,618 entries.

Valency-Based Features We use the valency dictionary to give us features based on selectional preferences. The more compatible a parse is with the selectional preferences the higher its score. The valency-based features (SP) were constructed by first finding the most appropriate pattern, and then recording how well it matched.

Figure 12 shows how we find the most appropriate pattern from the candidate dictionary entries whose lemma was the same as the predicate. As shown in Fig. 12, we adjust the compatibility score $score(N_j)$ according to the case-role (following Bond and Shirai 1997). This reflects the strength of the case-element's connection to the predicate. By these steps, we get both the most appropriate pattern Pat, the compatibility scores ($score(N_j)$s and $score(Pat)$), and matching relations (e.g. subsumption).

Once we have the most appropriate pattern, we then construct features that record how good the match is (Table 5). These include: the total score ($score(Pat)$), with or without the verb's Pattern ID (High/Med/Low/Zero: #31 0,1), the number of filled arguments (#31 2), the fraction of filled arguments vs all arguments (High/Med/Low/Zero: #31 3, 4), the score for each argument ($score(N_j)$) of the pattern (#33 5) and the types

Table 6 Data of Sets for Evaluation

Corpus		# Sents	Length(Average)	Parses/Sent(Average)
Definitions	Train	30,345	9.3	190.1
	Test	2,790	10.1	177.0
Examples	Train	27,081	10.9	74.1
	Test	2,587	10.4	47.3

of matches (**Child**) (#32 5, 7). We put the scores ($score(Pat)$ and $score(N_j)$) into four bins (High/Med/Low/Zero) to reduce sparseness.

These scores allow us to use information about word usage in an existing dictionary. We hope that this will provide a smoothing effect, adding information about selectional preferences for verb senses that did not occur in the training data.

4 Evaluation

We trained and tested using a subset of the dictionary definition and example sentences in the Hinoki corpus. This consists of those sentences with ambiguous parses that have been annotated so that the number of parses has been reduced (Table 6). That is, we excluded unambiguous sentences (with a single parse), and those where the annotators judged that no parse provided by JACY gave the correct semantics. This does not necessarily mean that there is a single correct parse; we allow the annotator to claim that two or more parses are equally appropriate. This left us with roughly half of the original corpus of 75,000 definitions and 45,000 examples.

Dictionary sentences are a different genre to other commonly used test sets (e.g. newspaper text in the Penn Treebank or tourism data in Redwoods). The main differences from newspaper text is that the definition sentences are shorter, contain more fragments (especially NPs as single utterances) and fewer quoted sentences and proper names. The main differences from travel dialogues is the lack of questions. However, they are valid examples of naturally occurring texts and a native speaker can read and understand them without special training. In addition, example sentences cover the typical usage of common words, and are therefore a good baseline for any genre.

4.1 Maximum Entropy Ranker

Log-linear models provide a very flexible framework that has been widely used for a range of tasks in NLP, including parse selection and reranking for machine translation. We use a *maximum entropy/minimum divergence* (MEMD) modeler to train the parse selection model. Specifically, we use the open-source **Toolkit for Advanced Discriminative Modeling** (TADM:[6] Malouf (2002) for training, using its *limited memory variable metric* as the optimization method and determining best-performing convergence

[6] http://tadm.sourceforge.net.

Table 7 Parse Selection Results (SYN, SEM and SYN-SEM)

Method	Definitions		Examples	
	Accuracy (%)	Features (×1000)	Accuracy (%)	Features (×1000)
SYN baseline	16.4	random	22.3	random
SYN-1	52.8	7	67.6	8
SYN-GP	62.7	266	76.0	196
SYN-ALL	**63.8**	316	**76.2**	245
SEM baseline	20.3	random	22.8	random
SEM-Dep	57.3	623	58.7	675
+SEM-WS	56.2	1,904	59.0	1,486
+SEM-Class	57.5	2,018	59.7	1,669
+SEM-L2	60.3	808	62.9	823
+SP	59.5	723	**68.2**	819
SEM-ALL1 (w/o SEM-WS)	**63.3**	1,713	68.9	1,868
SEM-ALL2 (w SEM-WS)	**63.3**	2,546	**69.2**	2,679
SYN-SEM1 (w/o SEM-WS)	69.5	2,029	**79.2**	2,126
SYN-SEM2 (w SEM-WS)	**69.8**	2,861	78.8	2,923

+ means something is added to SEM-Dep
SEM-ALL1 = SEM-Dep+ SEM-Class + SEM-L2 + SP
SEM-ALL2 = SEM-Dep+ SEM-Class + SEM-L2 + SP + SEM-WS

thresholds and prior sizes experimentally. A comparison of this learner with the use of support vector machines over similar data found that the SVMs gave comparable results but were far slower (Baldridge and Osborne 2007; Velldal 2008). Because we are investigating the effects of various different features, we chose the faster learner.

5 Results and Discussion

The results for most of the models discussed in the previous section are shown in Tables 7 and 8. The accuracy is an exact match for the entire sentence: a model gets a point only if its top-ranked analysis is the same as an analysis selected as correct in Hinoki. This is a stricter metric than component-based measures (e.g., labeled precision), which award partial credit for incorrect parses. For the syntactic models, the baseline (random choice) is 16.4% for the definitions and 22.3% for the examples. Definition sentences are harder to parse than example sentences. This is mainly because the examples have fewer relative clauses and coordinate NPs, which are both large sources of ambiguity. For the semantic and combined (SYN-SEM) models, multiple sentences can have different parses but the same semantics. In this case all sentences with the correct semantics are scored as good. This raises the baselines to 20.3 and 22.8% respectively.

Table 7 shows that even the simplest models (SYN-1 and SEM-Dep) provide a large improvement over the random baselines. With syntactic models, adding

Table 8 Parse Selection Results: Investigation of Superordinate Semantic Class Selection

Method	Definitions			Examples		
	Class No.	Accuracy (%)	Features (×1000)	Class No.	Accuracy (%)	Features (×1000)
SEM baseline	–	20.3	random	–	22.8	random
SEM-Dep	–	57.3	623	–	58.7	675
+SEM-Class	2,234	57.5	2,018	2,356	59.7	1,669
+SEM-L2 (9)	8	**60.3**	808	9	**62.9**	823
+SEM-L3 (30)	28	59.8	876	29	62.8	879
+SEM-L4 (136)	129	59.9	1,000	129	62.3	973
+SEM-L5 (392)	350	**60.4**	1,240	360	61.3	1,202
+SEM-VC500	392	59.8	1,473	317	60.2	1,552
+SEM-VC1000	228	60.0	1,328	172	60.3	1,385
+SEM-VC2000	120	60.1	1,167	92	61.2	1,237
+SEM-VC3000	72	60.1	1,059	58	62.0	1,151
+SEM-VC4000	58	**60.3**	1,026	45	61.5	1,117
+SEM-VC5000	48	59.4	998	35	61.5	1,079
+SEM-VC6000	40	59.3	974	30	61.5	1,055
+SEM-VC7000	32	59.1	946	24	61.7	1,029
+SEM-VC8000	26	59.3	919	20	61.9	1,011
+SEM-VC9000	24	59.4	908	18	62.3	1,005
+SEM-VC10000	23	59.5	906	14	62.4	984
+SEM-VC15000	15	59.5	867	11	62.0	968
+SEM-VC20000	11	59.6	840	9	61.9	950
+SEM-VC30000	8	59.8	825	8	61.9	949

grandparenting leads to a large improvement (SYN-GP). Adding lexical n-grams resulted in relatively slight improvement over this (SYN-ALL).

Overall, the semantic models achieve almost the same accuracy as the syntactic ones (although on a slightly easier task as there is some spurious syntactic ambiguity, which results in the semantic models having a higher random baseline). However, we had not expected that using **only** semantic features could work as well as syntactic features for parse selection, so this was a pleasing result. Moreover, the combined (SYN-SEM) models achieve the best result. The effect of smoothing with superordinate semantic classes (SEM-L), realizes a modest improvement over just the elementary dependencies. The features using the valency dictionary (SP) provide a considerable improvement especially for examples. The features using semantic classes (SEM-Class) provide a slight improvement. The features using word senses (SEM-WS) provide a limited improvement over examples, but result in a degradation over definitions, probably due to the sparseness. Combining all the semantic features (SEM-ALL1,2) provides a clear improvement, suggesting that the information is to some extent heterogeneous.

5.1 Analysis of the Effect of Semantic Features

In this section, we analyse the effects of adding the semantic information. We show an example where the semantic features gave a correct parse that the syntactic model missed.

(2) テニスの 試合 で 彼 と ペアを 組ん だ
tenisu no shiai de kare to pea wo kun da
tennis of match in him with pair ACC make past
"In a tennis match, I paired with him. (Lit: In tennis match, paired with him)"

The difficulty in (1) is that "I" is a zero pronoun. In the parse preferred by the syntactic model, と *to* "with" is interpreted as "and", that is "彼 *kare* "him" and ペア *pea* "pair"", thus is glossed as "(I) made him and a pair". But by using semantic classes, と *to* "with" is correctly interpreted as comitative case, glossed as "(I) made a pair with him".

Usually, conjunctions such as と *to* "and" and や *ya* "conjunction" list similar things, as shown in the definition of 運転手$_1$ *untenshu* "chauffeur"(see Fig. 1 and so on). In that sentence, the semantically close words, 電車 *densha* "train" and 自動車 *jidousha* "car" (⊂ ⟨**988:land vehicle**⟩), are conjoined. In contrast, 彼 *kare* "him" (⊂ ⟨**48:male/man**⟩ ⊂ ⟨**3:agent**⟩), and ペア *pea* "pair" (⊂ ⟨**2603:versus**⟩ ⊂ ⟨**2422:abstract relationship**⟩) are in very different semantic classes.

Another example is shown in (1). In this example, the syntactic model preferred the parse where 河童 *kappa* "cucumber roll" is interpreted as ARG1 of 抜く *nuku* "don't use" , and the gloss is "Cucumber roll don't use wasabi, please." But in the correct parse, it is the topic.

The semantic classes give the information that 河童 *kappa* "cucumber roll" is a kind of sushi and thus in ⟨**848:rice/cooked rice/meal/diet**⟩ (⊂ ⟨**533:objects**⟩). In general ⟨**533:objects**⟩ are not preferred as ARG1.

(3) 河童 は わさびを 抜い て ください
kappa ha wasabi wo nui te kudasai
cucumber roll TOP wasabi ACC don't use please
"Don't use wasabi for cucumber rolls, please."

As shown above, semantic information is effective in the problematic cases of postposition attachment ambiguity, coordinate structures and deciding whether something is an argument or adjunct.

5.2 Investigation of Appropriate Superordinate Semantic Classes

We investigate the most appropriate superordinate semantic classes in Table 8. SEM-LX shows the result using the semantic classes at a given level X. SEM-VCY shows the result using selected semantic classes based on its frequency (Y is the threshold). Surprisingly, as we add more data, the very top level of the semantic

Table 9 Distribution of Used Superordinate Semantic Classes: SEM-L2

Used Semantic Class	Lvl	Definitions		Examples	
		No.	(%)	No.	(%)
⟨**1:common noun**⟩	0	2,402	0.6	116	0
⟨**2:concrete**⟩	1	60	0	20	0
⟨**1000:abstract**⟩	1	2	0	2	0
⟨**3:agent**⟩	2	23,040	5.6	36,907	12.2
⟨**388:place**⟩	2	13,389	3.3	13,971	4.6
⟨**533:object**⟩	2	42,036	10.3	28,660	9.4
⟨**1001:abstract thing**⟩	2	28,511	7	17,422	5.7
⟨**1235:event**⟩	2	122,935	30.1	90,364	29.8
⟨**2422:abstract relationship**⟩	2	89,053	21.8	45,425	15.0
Total		408,507	100.0	303,452	100.0

class hierarchy performs almost as well as the more detailed levels. For the definitions, SEM-L5 gives the best result, but SEM-L2 was very close. For the examples, SEM-L2 gives the best result. Frequency-based superordinate semantic classes (SEM-VC) provides comparable results, but they were not the best. Thus, it should be concluded that the superordinate semantic classes at level 2 (SEM-L2) are the best for this task.

To give an idea of the sense granularity we found useful, we show the distribution of semantic classes for SEM-L2 in Table 9. The definitions and examples have a similar distribution. The main differences are the ratio of ⟨**3:agent**⟩ and ⟨**2422:abstract relationship**⟩ (±6.6 – 6.8%).

The use of hand-crafted lexical resources such as the Goi-Taikei ontology is sometimes criticized on the grounds that such resources are hard to produce and thus scarce. While it is true that valency lexicons and sense hierarchies are hard to produce, they are of such value that they have already been created for all of the languages we know of that have large treebanks. In fact, there are more languages with WordNets than large treebanks. Therefore, there is no principled reason not to use hand built semantic reources.

5.3 Learning Curves

Finally, as shown in Table 7, combining the syntactic and semantic features gives the best results by far (SYN-SEM: SYN-ALL + SEM-ALL1 or 2). The definition sentences are harder syntactically, and thus get more of a boost from the semantics. The semantics still improves the performance for the example sentences.

The semantic-class-based sense features used here are based on manual annotation, and thus reveal an upper bound for the effects of these features. This is not an absolute upper bound for the use of sense information—it may be possible to archive further improvement through feature engineering. In addition, as the learning curves (Fig. 12) have not yet flattened out, there is still room to improve the results by increasing the size of the training data.

6 Related Work

Bikel (2000) combined sense information and parse information using a subset of SemCor (with WordNet senses and Penn-II Treebanks) to produce a combined model. This model did not outperform a purely syntactic model. It did not use semantic dependency relations, but only syntactic dependencies augmented with heads, which suggests that the deeper structural semantics provided by the HPSG parser is important. Xiong et al. (2005) achieved a minor improvement over a plain syntactic model, using features based on both the correlation between predicates and their arguments, and between predicates and the hypernyms of their arguments (using HowNet Dong and Dong (2000)). However, they did not investigate generalizing to levels other than a word's immediate hypernym. Recently, Agirre et al. (2008) showed that semantic classes help to obtain significant improvements in both parsing and PP attachment tasks using models similar to ours. Their tests employed English datasets: Penn Treebank, SemCor and WordNet. Interestingly, they found that automatically disambiguated sense information was as effective as the gold standard information.

Pioneering work by Toutanova et al. (2005) and Baldridge and Osborne (2007) on parse selection for an English HPSG treebank used simpler semantic features without sense information, and they obtained a far less dramatic improvement when they combined syntactic and semantic information.

Miyao and Tsujii (2008) use an efficient feature forest model for HPSG parse selection. They compared syntactic features (comparable to our SYN-1) and semantic features (comparable to our SEM-Dep). They also found that for the simple models syntactic models outperformed semantic models (34% sentence accuracy vs. 28.9% on the Penn Treebank). This is not so surprising because word level dependency features are very sparse, and without some kind of sense disambiguation it is impossible to back-off. Because this corpus has not been sense annotated, they were unable to test more sophisticated semantic models.

Looking at various attempts to use semantic features, we conclude that it isn't enough to just add senses or semantic dependencies in as features, there has to be some kind of generalization to hypernyms to make things work. We look on the higher level semantic classes as analogous to part of speech classes in syntactic modeling—they allow us to capture useful generalizations without the data becoming too sparse.

7 Conclusion

In summary, we showed that both syntactic and semantic features are effective in ranking parses. Sense-based features do best when combined with a semantic hierarchy and generalized to superordinate senses. Adding information about selectional preferences also improves performance. When training and testing on the definition subset of the Hinoki corpus, a combined model achieved a 5.6% improvement in parse selection accuracy over a model using only syntactic features (63.8% → 69.8%). Similar results (76.2% → 79.2%) were obtained for example sentences.[7]

These general results have been shown to hold true for English (Agirre et al. 2008), and we hope will lead to more research combining syntax and semantics in parsing, as well as renewed interest in producing richly annotated corpora. In future work we intend to confirm that we can obtain improved results with automatically disambiguated senses, not just with the gold standard annotations, and test the results on other sections of the Hinoki corpus, such as newspaper articles.

Acknowledgement The authors would like to thank the other members of the NTT Machine Translation Research Group, as well as Timothy Baldwin, Dan Flickinger, Erik Velldal and the various reviewers who have commented on and encouraged this paper. This research was supported by the research collaboration between the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation, and the Center for the Study of Language and Information (CSLI) at Stanford University, through the research partnership between the University of Oslo and CSLI and through the joint research between NTT Communication Science Laboratories and the National Institute for Information and Communications Technologies (NICT), Japan.

References

Abney, S. P. (1997). Stochastic attribute-value grammars. *Computational Linguistics, 23*, 597–618.

Agirre, E., Baldwin, T., & Martinez, D. (2008). Improving parsing and PP attachment performance with sense information. In *Proceedings of the 46th annual meeting of the association for computational linguistics: ACL/HLT-2008* (pp. 317–325).

Baldridge, J., & Osborne, M. (2007). Active learning and logarithmic opinion pools for HPSG parse selection. *Natural Language Engineering, 13*(1), 1–32.

Bikel, D. M. (2000). A statistical model for parsing and word-sense disambiguation. In *ACL-2000 student research workshop* (pp. 1–7). Hong Kong.

Bond, F., Fujita, S., & Tanaka, T. (2006). The Hinoki syntactic and semantic treebank of Japanese. *Language Resources and Evaluation, 40*(3–4), 253–261. (Special issue on Asian language technology).

Bond, F., & Shirai, S. (1997). Practical and efficient organization of a large valency dictionary. In *NLPRS-97 Workshop on Multilingual Information Processing*. Phuket.

Ciaramita, M., & Altun, Y. (2006). Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 conference on empirical methods in natural language processing: EMNLP-2006* (pp. 594–60)2. Sydney, Australia.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal recursion semantics. An Introduction. *Research on Language and Computation, 3*(4), 281–332.

Dong, Z., & Dong, Q. (2000). http://www.keenage.com.

Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.

Fujita, S., & Bond, F. (2007). A method of creating new valency entries. *Machine Translation Journal, 21*(1), 1–28.

Fujita, S., Bond, F., Oepen, S. & Tanaka, T. (2007). Exploiting semantic information for HPSG parse selection. In *Proceedings of ACL-2007 workshop on deep linguistic processing* (pp. 25–32). Prague, Czech Republic.

Ginzburg, J., & Sag, I. A. (2000). *Interrogative investigations. The form, meaning, and use of English interrogatives*. Stanford, CA: CSLI Publications.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). OntoNotes: The 90% Solution. In *Proceedings of the human language technology conference of the NAACL, companion volume: short papers* (pp. 57–60). New York City, USA: Association for Computational Linguistics

[7] Note that the baseline of the combined models is higher than that of syntactic models. All accuracy results are for sentence accuracy.

Ikehara, S., Miyazaki, M., Shirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y., & Hayashi, Y. (1997). *Goi-Taikei—A Japanese Lexicon*. Iwanami Shoten, Tokyo. 5 volumes/CD-ROM.

Johnson, M., Geman, S., Canon, S., Chi, Z. & Riezler, S. (1999). Estimators for Stochastic 'Unification-based' Grammars. In *Proceedings of the 37th annual meeting of the association for computational linguistics: ACL-99* (pp. 535–541). College Park, MD.

Kasahara, K., Sato, H., Bond, F., Tanaka, T., Fujita, S., Kanasugi, T., & Amano, S. (2004). Construction of a Japanese semantic Lexicon: Lexeed. In *IEICE technical report: 2004-NLC-159*, pp. 75–82. (in Japanese).

Kindaichi, H., & Ikeda, Y. (1988). *Gakken Japanese Dictionary* (2nd ed.). Tokyo, Japan: Gakken Co Ltd.

Klein, D. & Manning, C. D. (2003). Accurate unlexicalized parsing. In Erhard, H. & Dan R. (Eds.), *Proceedings of the 41st annual meeting of the association for computational linguistics*, (pp. 423–430).

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th conference on computational natural language learning: CoNLL-2002*. Taipei, Taiwan.

Malouf, R., & van Noord, G. (2004). Wide coverage parsing with stochastic attribute value grammars. In *IJCNLP-04 Workshop: Beyond shallow analyses—Formalisms and statistical modeling for deep analyses.* JST CREST.

Miyao, Y., & Tsujii, J. (2008). Feature forest models for probabilistic HPSG parsing. *Computational Linguistics, 34*(1), 35–80.

Oepen, S., Flickinger, D., Toutanova, K., & Manning, C. D. (2004). LinGO Redwoods: A rich and dynamic treebank for HPSG. *Research on Language and Computation, 2*(4), 575–596.

Oepen, S. & Lønning, J. T. (2006). Discriminant-based MRS banking. In *Proceedings of the 5th international conference on language resources and evaluation: LREC-2006*. Genoa, Italy.

Pollard, C., & Sag, I. A. (1994). *Head driven phrase structure grammar*. Chicago: University of Chicago Press.

Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell III, J. T., Alto, P. & Johnson, M. (2002). Parsing the Wall Street Journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of the 40th annual meeting of the association for computational linguistics: ACL-2002*. Philadelphia, PA.

Siegel, M., & Bender, E. M. (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.

Toutanova, K., Manning, C. D., Flickinger, D., & Oepen, S. (2005). Stochastic HPSG parse disambiguation using the redwoods corpus. *Research on Language and Computation, 3*(1), 83–105.

Velldal, E. (2008). *Empirical realization ranking*. Doctoral dissertation, University of Oslo.

Xiong, D., Li, S., Liu, Q., Lin, S., & Qian, Y. (2005). *Parsing the Penn Chinese treebank with semantic knowledge*. In Dale, R., Su, J., Wong, K.-F., & Kwong, O. Y. (Eds.), *Natural language processing—IJCNLP 005: Second international joint conference proceedings*, (pp. 70–81). Springer.