<table>
<thead>
<tr>
<th>Title</th>
<th>Word sense disambiguation incorporating lexical and structural semantic information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Tanaka, Takaaki; Bond, Francis; Baldwin, Timothy; Fujita, Sanae; Hashimoto, Chikara</td>
</tr>
<tr>
<td>Date</td>
<td>2007</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/6449">http://hdl.handle.net/10220/6449</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 2007 ACL This is the author created version of a work that has been peer reviewed and accepted for publication by Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). Association for Computational Linguistics. It incorporates referee’s comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [URL: <a href="http://www.aclweb.org/anthology-new/D/D07/D07-1050.pdf">http://www.aclweb.org/anthology-new/D/D07/D07-1050.pdf</a>].</td>
</tr>
</tbody>
</table>
Word Sense Disambiguation
Incorporating Lexical and Structural Semantic Information

Takaaki Tanaka† Francis Bond♦ Timothy Baldwin♠ Sanae Fujita† Chikara Hashimoto♣
† {takaaki, sanae}@cslab.kecl.ntt.co.jp ♦ bond@nict.go.jp
♠ tim@csse.unimelb.edu.au ♣ ch@yz.yamagata-u.ac.jp

† NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation
♦ National Institute of Information and Communications Technology
♠ The University of Melbourne ♣ Yamagata University

Abstract

We present results that show that incorporating lexical and structural semantic information is effective for word sense disambiguation. We evaluated the method by using precise information from a large treebank and an ontology automatically created from dictionary sentences. Exploiting rich semantic and structural information improves precision 2–3%. The most gains are seen with verbs, with an improvement of 5.7% over a model using only bag of words and n-gram features.

1 Introduction

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 being combined with stochastic models (Riezler et al., 2002; Oepen et al., 2002; Malouf and van Noord, 2004). Statistical techniques have also been shown to be useful for word sense disambiguation (Stevenson, 2003). However, to date, there have been few 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. Features based on general relations provide little improvement, 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 (⊃ stocks) and move upward (⊃ skyrocket) frequently appear together. However, there has been little success in this area to date. For example, Xiong et al. (2005) use semantic knowledge to parse Chinese, but gain only a marginal improvement. Focusing on WSD, Stevenson (2003) and others have shown that the use of syntactic information (predicate-argument relations) improve the quality of word sense disambiguation (WSD). McCarthy and Carroll (2003) have shown the effectiveness of the selectional preference information for WSD. However, there is still little work on combining WSD and parse selection.

We hypothesize that one of the reasons for the lack of success is that there has been no resource annotated with both syntactic (or structural semantic information) and lexical semantic information. For English, there is the SemCor corpus (Fellbaum, 1998) which is annotated with parse trees and WordNet senses, but it is fairly small, and does not explicitly include any structural semantic information. Therefore, we decided to construct and use a treebank with both syntactic information (e.g. HPSG parses) and lexical semantic information (e.g. sense tags): the Hinoki treebank (Bond et al., 2004). This can be used to train word sense disambiguation and parse ranking models using both syntactic and lexical semantic features. In this paper, we discuss only word sense disambiguation. Parse ranking is discussed in Fujita et al. (2007).

2 The Hinoki Corpus

The Hinoki corpus consists of the Lexeed Semantic Database of Japanese (Kasahara et al., 2004) and corpora annotated with syntactic and semantic infor-
2.1 Lexeed
Lexeed is a database built from a dictionary, which defines word senses used in the Hinoki corpus and has around 49,000 dictionary definition sentences and 46,000 example sentences which are syntactically and semantically annotated. Lexeed consists of all words with a familiarity greater than or equal to five on a scale of one to seven. This gives a fundamental vocabulary of 28,000 words, divided into 46,347 different senses. Each sense has a definition sentence and example sentence written using only these 28,000 familiar words (and some function words). Many senses have more than one sentence in the definition: there are 75,000 defining sentences in all.

A (simplified) example of the entry for 運転手 "chauffeur" is given in Figure 1. Each word contains the word itself, its part of speech (POS) and lexical type(s) in the grammar, and the familiarity score. Each sense then contains definition and example sentences written using only these 28,000 familiar words (and some function words). Many senses have more than one sentence in the definition: there are 75,000 defining sentences in all.

A (simplified) example of the entry for 運転手 "chauffeur" is given in Figure 1. Each word contains the word itself, its part of speech (POS) and lexical type(s) in the grammar, and the familiarity score. 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 (Ikehara et al., 1997) and WordNet (Fellbaum, 1998). Each content word in the definition and example sentences is annotated with sense tags from the same lexicon.

2.2 Lexical Semantics Annotation
The lexical semantic annotation uses the sense inventory from Lexeed. All words in the fundamental vocabulary are tagged with their sense. For example, the word 大きい ookii “big” (in ookiku naru “grow up”) is tagged as sense 5 in the example sentence (Figure 1), with the meaning “elder, older”.

Each word was annotated by five annotators. We use the majority choice in case of disagreements (Tanaka et al., 2006). Inter-annotator agreements among the five annotators range from 78.7% to 83.3%: the lowest agreement is for the Lexeed definition sentences and the highest is for Kyoto corpus (newspaper text). These agreements reflect the difficulties in disambiguating word sense over each corpus and can be considered as the upper bound of precision for WSD.

Table 1 shows the distribution of word senses according to the word familiarity in Lexeed.

<table>
<thead>
<tr>
<th>Fam</th>
<th>#Words</th>
<th>Polysemous</th>
<th>#WS</th>
<th>#Monosemous(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5</td>
<td>368</td>
<td>182</td>
<td>4.0</td>
<td>186 (50.5)</td>
</tr>
<tr>
<td>6.0</td>
<td>4,445</td>
<td>1,902</td>
<td>3.4</td>
<td>2,543 (57.2)</td>
</tr>
<tr>
<td>5.5</td>
<td>9,814</td>
<td>3,502</td>
<td>2.7</td>
<td>6,312 (64.3)</td>
</tr>
<tr>
<td>5.0</td>
<td>11,430</td>
<td>3,457</td>
<td>2.5</td>
<td>7,973 (69.8)</td>
</tr>
</tbody>
</table>

Table 1: Word Senses in Lexeed

2.3 Ontology
The Hinoki corpus comes with an ontology semi-automatically constructed from the parse results of definitions in Lexeed (Nichols and Bond, 2005). The ontology includes more than 80 thousand relationships between word senses, e.g. synonym, hypernym, abbreviation, etc. The hypernym relation for 運転手 untenshu “chauffeur” is shown in Figure 1. Hypernym or synonym relations exist for almost all content words.

2.4 Thesaurus
As part of the ontology verification, all nominal and most verbal word senses in Lexeed were linked to semantic classes in the Japanese thesaurus, Nihongo Goi-Taikei (Ikehara et al., 1997). These were then hand verified. Goi-Taikei has about 400,000 words including proper nouns, most nouns are classified into about 2,700 semantic classes. These semantic classes are arranged in a hierarchical structure (11 levels). The Goi-Taikei Semantic Class for 運転手 untenshu “chauffeur” is shown in Figure 1: ⟨C292:driver⟩ at level 9 which is subordinate to ⟨C4:person⟩.

2.5 Syntactic and Structural Semantics Annotation
Syntactic annotation is done by selecting the best parse (or parses) from the full analyses derived by a broad-coverage precision 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) it is possible to simultaneously annotate syntactic and semantic structure without overburdening the annotator. Using a grammar enforces treebank consistency — all sentences annotated are guaranteed to have well-
formed parses. The flip side to this is that any sentences which the parser cannot parse remain unannotated, at least unless we were to fall back on full manual mark-up of their analyses. The actual annotation process uses the same tools as the Redwoods treebank of English (Oepen et al., 2002).

There were 4 parses for the definition sentence shown in Figure 1. The correct parse, shown as a phrase structure tree, is shown in Figure 2. The two sources of ambiguity are the conjunction and the relative clause. The parser also allows the conjunction to join to 電車 densha and 人 hito. In Japanese, relative clauses can have gapped and non-gapped readings. In the gapped reading (selected here), 人 hito is the subject of 運転 unten “drive”. In the non-gapped reading there is some underspecified relation between the thing 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). Such semantic ambiguity is resolved by selecting the correct derivation tree that includes the applied rules in building the tree.

The parse results can be automatically given by the HPSG parser PET (Callmeier, 2000) with the Japanese grammar JACY. The current parse ranking model has an accuracy of 70%: the correct tree is ranked first 70% of the time (for Lexeed definition sentences) (Fujita et al., 2007).

The full parse is an HPSG sign, containing both syntactic and semantic information. A view of the semantic information is given in Figure 3. The specific meaning representation language used in JACY is Minimal Recursion Semantics (Copestake et al., 2005).
3 Task

We define the task in this paper as “allocating the word sense tags to all content words included in Lexeed as headwords, in each input sentence”. This task is a kind of all-words task, however, a unique point is that we focus on fundamental vocabulary (basic words) in Lexeed and ignore other words. We use Lexeed as the sense inventory. There are two problems in resolving the task: how to build the model and how to assign the word sense by using the model for disambiguating the senses. We describe the word sense selection model we use in section 4 and the method of word sense assignment in section 5.

4 Word Sense Selection Model

All content words (i.e. basic words) in Lexeed are classified into six groups by part-of-speech: noun, verb, verbal noun, adjective, adverb, others. We treat the first five groups as targets of disambiguating senses. We build five words sense models corresponding to these groups. A model contains senses for various words, however, features for a word are discriminated from those for other words so that the senses irrelevant to a target word are not selected. For example, an n-gram feature following a target word “has-a-tail” for dog is distinct from that for cat.

In the remainder of this section, we describe the features used in the word sense disambiguation. First we used simple n-gram collocations, then a bag of words of all words occurring in the sentence. This was then enhanced by using ontological information and predicate argument relations.

4.1 Word Collocations

Word collocations (WORD-Col) are basic and effective cues for WSD. They can be modelled by n-gram and bag of words features, which are easily extracted from a corpus. We used all unigrams, bigrams and trigrams which precede and follow the target words (N-gram) and all content words in the sentences where the target words occur (BOW).
# sample features

C1 ⟨COL WS:/BC
4 ⟩
C2 ⟨COL WSSC:C33:other person⟩
C3 ⟨COL WSHYP:/BC /BB
1 ⟩
C4 ⟨COL ... used the sense information from the corpus combined with the semantic classes in the dictionary to replace each pred-

4.2 Semantic Features

We use the semantic information (sense tags and ontologies) in two ways. One is to enhance the collocations and the other is to enhance dependency relations.

4.2.1 Semantic Collocations

Word surface features like N-gram and BOW inevitably suffer from data sparseness, therefore, we generalize them to more abstract words or concepts and also consider words having the same meanings. We used the ontology described in Section 2.3 to get hypernyms and synonyms and the Goi-Taikei thesaurus to abstract the words to the semantic classes. The superordinate classes at level 3, 4 and 5 are also added in addition to the original semantic class. For example, 電車 densha “train” and 自動車 jidousha “automobile” are both generalized to the semantic class ⟨C988:land vehicle⟩ (level 7). The superordinate classes are also used: ⟨C706:inanimate⟩ (level 3), ⟨C760:artifact⟩ (level 4) and ⟨C988:vehicle⟩ (level 5).

4.2.2 Semantic Dependencies

The semantic dependency features are based on a predicate and its arguments taken from the elementary dependencies. For example, consider the semantic dependency representation for densha ya

<table>
<thead>
<tr>
<th>#</th>
<th>sample features for 運転する</th>
<th>sample features</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 (PRED:運転する, ARG1:人)</td>
<td>D1 (PRED:運転する, ARG1:人)</td>
<td></td>
</tr>
<tr>
<td>C1 (PRED:運転する, ARG2:電車)</td>
<td>D1 (PRED:運転する, ARG2:自動車)</td>
<td></td>
</tr>
<tr>
<td>C2 (PRED:運転する, ARG1:人)</td>
<td>D2 (PRED:運転する, ARG2:電車)</td>
<td></td>
</tr>
<tr>
<td>C2 (PRED:運転する, ARG2:電車)</td>
<td>D2 (PRED:運転する, ARG2:自動車)</td>
<td></td>
</tr>
<tr>
<td>C3 (PRED:運転する, ARG1:C33)</td>
<td>D3 (PRED:運転する, ARG2:C988)</td>
<td></td>
</tr>
<tr>
<td>C5 (PRED:運動する, ARG1:C5, ARG2:電車)</td>
<td>D5 (PRED:運動する, ARG2:C988)</td>
<td></td>
</tr>
<tr>
<td>D22 (PRED:運動する, ARG1:人, ARG2:電車)</td>
<td>D23 (PRED:運動する, ARG1:人, ARG2:C1460)</td>
<td></td>
</tr>
<tr>
<td>D33 (PRED:運動する, ARG1:C5, ARG2:C988)</td>
<td>D33 (PRED:運動する, ARG1:C5, ARG2:C988)</td>
<td></td>
</tr>
<tr>
<td>D56 (PRED:運動する, ARG1:C5, ARG2:電車)</td>
<td>D65 (PRED:運動する, ARG1:C5, ARG2:電車)</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Example semantic collocation features (SEM-Col) extracted from the word sense tagged corpus and the dictionary (Lexeed and GoiTaikei) and the ontology which have the word senses and the semantic classes linked to the semantic tags. The first column numbers the feature template corresponding to each example.

Table 3: Example semantic features extracted from the dependency tree in Figure 4. The first column numbers the feature template corresponding to each example.

jidousha-wo unten suru hito “a person who drives a train or car” given in Figure 4. The predicate unten “drive”, has two arguments: ARG1 hito “person” and ARG2 ya “or”. The coordinate conjunction is expanded out into its children, giving ARG2 densha “train” and jidousha “automobile”.

From these, we produce several features, a sample of them are shown in Table 3. One has all arguments and their labels (D11). We also produce various back offs, for example the predicate with only one argument at a time (D1-D3). Each combination of predicate and its related argument(s) becomes a feature.

For the next class of features, we used the sense information from the corpus combined with the semantic classes in the dictionary to replace each pred-
icate by its disambiguated sense, its hypernym, its synonym (if any) and its semantic class. The semantic classes for 電車 和 自動車 are both (988:1 and vehicle), while 运転 is (2003:motion) and 人 is (4:human). We also expand 自動車 into its synonym モーターカー mōtākā “motor car”.

The semantic class features provide a semantic smoothing, as words are binned into the 2,700 classes. The hypernym/synonym features provide even more smoothing. Both have the effect of making more training data available for the disambiguator.

4.3 Domain

Domain information is a simple and sometimes strong cue for disambiguating the target words (Gliozzo et al., 2005). For instance, the sense of the word “record” is likely to be different in the musical context, which is recalled by domain-specific words like “orchestra”, “guitar”, than in the sporting context. We use 12 domain categories like “culture/art”, “sport”, etc. which are similar to ones used in directory search web sites. About 6,000 words are automatically classified into one of 12 domain categories by distributions in web sites (Hashimoto and Kurohashi, 2007) and 10% of them are manually checked. Polysemous words which belong to multiple domains and neutral words are not classified into any domain.

5 Search Algorithm

The conditional probability of the word sense for each word is given by the word sense selection model described in Section 4. In the initial state, some of the semantic features, e.g. semantic collocations (SEM-Col) and word sense extensions for semantic dependencies (SEM-Dep) are not available, since no word senses for polysemous words have been determined. It is not practical to count all combinations of word senses for target words, therefore, we first try to decide the sense for that word which is most plausible among all the ambiguous words, then, disambiguate the next word by using the sense.

We use the beam search algorithm, which is similar to that used for decoder in statistical machine translation (Watanabe, 2004), for finding the plausible combination of word sense tags.

The algorithm is described as follows. For a polysemous word set in an input sentence \( \{w_1, \ldots, w_n\} \), \( t_{w:k} \) is the \( k \)-th word sense of word \( w_i \), \( W \) is a set having words to be disambiguated, \( T \) is a list of resolved word senses. A search node \( N \) is defined as \([W,T]\) and a score of a node \( N \), \( s(N) \) is defined as the probability that the word sense set \( T \) occurs in the context. The beam search can be done as follows (beam width is \( b \)):

1. Create an initial node \( N_0 = [T_0, W_0]\) (\( T_0 = \{\}\), \( W_0 = \{\}\)) and insert the node into an initial queue \( Q_0 \).

2. For each node \( N \) in the queue \( Q \), do the following steps.
   - For each \( w_i \in W \), create \( W_i' \) by picking out \( w_i \) from \( W \)
   - Create new lists \( T_i' \) by adding one of word sense candidates \( t_i, \ldots, t_i/l \) for \( w_i \) to \( T \)
   - Create new nodes \([W_i', T_i], \ldots, [W_i', T'_i]\) and insert them into the queue \( Q_i \)

3. Sort the nodes in \( Q_i \) by the score \( s(N) \)

4. If the top node \( W \) in the queue \( Q_i \) is empty, adopt \( T \) as the combination of word senses and terminate. Otherwise, pick out the top \( b \) nodes from \( Q_i \) and insert them into new queue \( Q \), then go back to 2

6 Evaluation

We trained and tested on the Lexeed Dictionary Definition (LXD-DEF) and Example sections (LXD-EX) of the Hinoki corpus (Bond et al., 2007). These have about 75,000 definition and 46,000 example sentences respectively. Some 54,000 and 36,000 sentences of them are treebanked, i.e., they have the syntactic trees and structural semantic information. We used these sentences with the complete information and selected 1,000 sentences out of each sentence class as test sets (LXD-DEF\text{test}, LXD-EX\text{test}), and the remainder is combined and used as a training set (LXD-ALL). We also tested 1,000 sentences from the Kyoto Corpus of newspaper text (KYOTO\text{test}). These sentences have between 3.4 (LXD-EX\text{test}) – 5.2 (KYOTO\text{test}) polysemous words per sentence on average.
We use a maximum entropy / minimum divergence (MEMD) modeler to train the word sense selection model. We use the open-source Maximum Entropy Modeling Toolkit\(^2\) for training, determining best-performing convergence thresholds and prior sizes experimentally. The models for five different POSs were trained with each training sets: the base model is word collocation model (WORD-Col), and the semantic models built by semantic collocation (SEM-Col), semantic dependency (SEM-Dep) or domain with WORD-Col (+SEM-Col, +SEM-Dep and +DOMAIN).

Table 4 shows the precision as the results of the word sense disambiguation on the combination of LXD-DEF and LXD-EX (LXD-ALL). The baseline method selects the senses occurring most frequently in the training corpus. Each row indicates the results using the baseline, word collocation (WORD-Col), the combinations of WORD-Col and one of the semantic features (+SEM-Col, +SEM-Dep and +DOMAIN), e.g., +SEM-Col gives the results using WORD-Col and SEM-Col, and all features (FULL).

There are significant improvements over the baseline and the other results on all corpora. Basic word collocation features (WORD-Col) give a vast improvement. Extending this by using the ontological information (+SEM-Col) gives a further improvement over the WORD-Col. Adding the predicate-argument relationships (+SEM-Dep) improves the results even more.

Table 6 shows the statistics of the target corpora. The best result of LXD-DEF\(_{\text{test}}\) (80.7%) surpasses the inter-annotator agreement (78.7%) in building the Hinoki Sensebank. However, there is a wide gap between the best results of KYOTO\(_{\text{test}}\) (60.4%) and the inter-annotator agreement (83.3%), this suggests other information such as the semantic classes for named entities (including proper nouns and multi-word expressions (MWE)) and broader contexts are required. However, a model built on dictionary sentences lacks these features. Even so, there is some improvement.

The domain features (+DOMAIN) give small contribution to the precision, since only intra-sentence context is counted in this experiment. Unfortunately dictionary definition and example sentences do not really have a useful context. We expect broader context should make the domain features more effective for the newspaper text (e.g. as in Stevenson (2003)).

Table 5 shows comparison of results of different POSs. The semantic features (+SEM-Col and +SEM-Dep) are particularly effective for verb and also give moderate improvements on the results of the other POSs.

Figure 5 shows the precisions of LXD-DEF\(_{\text{test}}\) in changing the size of a training corpus, which is divided into five partitions. The precision is saturated in using four partitions (264,000 tokens).

These results of the dictionary sentences are close to the best published results for the SENSEVAL-2 task (79.3% by Murata et al. (2003) using a combination of simple Bayes learners). However, we are using a different sense inventory (Lexeed not Iwanami (Nishio et al., 1994)) and testing over a different corpus, so the results are not directly comparable. In future work, we will test over SENSEVAL-2 data so that we can compare directly.

None of the SENSEVAL-2 systems used ontological information, despite the fact that the dictionary definition sentences were made available, and there are several algorithms describing how to extract such information from MRDs (Tsurumaru

---

\(^2\)http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html
et al., 1991; Wilkes et al., 1996; Nichols et al., 2005). We hypothesize that this is partly due to the way the task is presented: there was not enough time to extract and debug an ontology as well as build a disambiguation system, and there was no ontology distributed. The CRL system (Murata et al., 2003) used a syntactic dependency parser as one source of features (KNP: Kurohashi and Nagao (2003)), removing it decreased performance by around 0.6%.

8 Conclusions

We used the Hinoki corpus to test the importance of lexical and structural information in word sense disambiguation. We found that basic n-gram features and collocations provided a great deal of useful information, but that better results could be gained by using ontological information and semantic dependencies.

Acknowledgements

We would like to thank the other members of the NTT Natural Language Research Group NTT Communication Science laboratories for their support. We would also like to express gratitude to the reviewers for their valuable comments and Professor Zeng Guangping, Wang Daliang and Shen Bin of the University of Science and Technology Beijing (USTB) for building the demo system.

References


<table>
<thead>
<tr>
<th>Corpus</th>
<th>Annotated Tokens</th>
<th>#WS</th>
<th>Agreement token (type)</th>
<th>%Other Sense</th>
<th>%Homonym</th>
<th>%MWE</th>
<th>%Proper Noun</th>
</tr>
</thead>
<tbody>
<tr>
<td>LXD-DEF</td>
<td>199,268</td>
<td>5.18</td>
<td>.787 (0.850)</td>
<td>4.2</td>
<td>0.084</td>
<td>1.5</td>
<td>0.046</td>
</tr>
<tr>
<td>LXD-EX</td>
<td>126,966</td>
<td>5.00</td>
<td>.820 (0.871)</td>
<td>2.3</td>
<td>0.035</td>
<td>0.4</td>
<td>0.0018</td>
</tr>
<tr>
<td>KYOTO</td>
<td>268,597</td>
<td>3.93</td>
<td>.833 (0.828)</td>
<td>9.8</td>
<td>3.3</td>
<td>7.9</td>
<td>5.5</td>
</tr>
</tbody>
</table>

Table 6: Corpus Statistics


485