Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

On Contribution of Sense Dependencies to Word Sense Disambiguation

Jun Hatori[†], Yusuke Miyao[†] and Jun'ichi Tsujii[†,††,†††]

Traditionally, many researchers have addressed word sense disambiguation (WSD) as an independent classification problem for each word in a sentence. However, the problem with their approaches is that they disregard the interdependencies of word senses. Additionally, since they construct an individual sense classifier for each word, their method is limited in its applicability to the word senses for which training instances are served. In this paper, we propose a supervised WSD model based on the syntactic dependencies of word senses. In particular, we assume that strong dependencies between the sense of a syntactic head and those of its dependents exist. We describe these dependencies on the tree-structured conditional random fields (T-CRFs), and obtain the most appropriate assignment of senses optimized over the sentence. Furthermore, we incorporate these sense dependencies in combination with various coarse-grained sense tag sets, which are expected to relieve the data sparseness problem, and enable our model to work even for words that do not appear in the training data. In experiments, we display the appropriateness of considering the syntactic dependencies of senses, as well as the improvements by the use of coarse-grained tag sets. The performance of our model is shown to be comparable to those of state-of-the-art WSD systems. We also present an in-depth analysis of the effectiveness of the sense dependency features by showing intuitive examples.

Key Words: *word sense disambiguation, tree-structured CRF, sense dependency*

1 Introduction

Word sense disambiguation (WSD) is one of the fundamental problems in computational linguistics. The task of WSD is to resolve the inherent polysemy of words by determining the appropriate sense(s) for each polysemous word in a given text. It is considered to be an intermediate, but necessary step for many NLP applications, including machine translation and information extraction, which require the knowledge of word senses to perform better.

One major obstacle for large-scale and precise WSD is the data sparseness problem caused by the fine-grained nature of the sense distinction. In recent years, in order to resolve this problem, several semi-supervised approaches have been explored. While some researchers have addressed

[†] Graduate School of Information Science and Technology, University of Tokyo
[††] School of Computer Science, University of Manchester
[†††] National Centre for Text Mining, UK

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

the scarcity of the training data directly, by exploring the methods to obtain more tagged instances from unannotated corpora (e.g. (Mihalcea 2004)), other researchers have used unannotated corpora to extract useful *global* information, such as the domain information (Gliozzo, Giuliano, and Strapparava 2005; Boyd-Graber, Blei, and Zhu 2007), and incorporated this information into supervised WSD frameworks. The use of *global* information extracted from unannotated corpora has succeeded in dramatically increasing the performance of WSD; however, on the other hand, the effectiveness of *local* or *syntactic* information has not been fully examined.

One such information yet to be explored is the interdependency of word senses. Although the use of local and syntactic information has been common in WSD, traditional approaches to supervised WSD are typically based on the individual classification framework for each word (Hoste, Hendrickx, Daelemans, and van den Bosch 2002; Decadt, Hoste, Daelemans, and den Bosch 2004), in which each word's sense is treated independently, regardless of any interdependencies or cooccurrences of word senses. Accordingly, the resulting sense assignment may be semantically inconsistent over the sentence. To solve this problem is of great interest from both a practical and theoretical viewpoint.

In this paper, we present a WSD model that naturally handles all content words in a sentence. We focus on using the interdependency of word senses, so that our model can output a semantically consistent assignment of senses to the whole sentence. Specifically, we assume that there are strong sense dependencies between a syntactic head and its dependents in the dependency tree. Furthermore, we combine these sense dependencies with various coarse-grained sense tag sets. These combined features are expected to alleviate the data sparseness problem, and also enable our model to work even for words that do not appear in the training data, which traditional individual classifiers cannot handle.

As a machine learning method, we adopt the tree-structured conditional random fields (T-CRFs) (Tang, Hong, Li, and Liang 2006). We solve WSD as a labeling problem to a sentence described as a dependency tree, where the vertices correspond to the words, and the edges correspond to the sense dependencies. T-CRFs also enable us to incorporate various sense tag sets all together into a simple framework.

In our experiments, three interesting results are found: the interdependency of word senses contribute to the improvement of WSD models, the combined features with coarse-grained sense tags work effectively, and the tree-structured model outperforms the linear-chain model. These results are confirmed on three data sets (the SemCor corpus and the SENSEVAL-2 and -3 English all-words task test sets) and on two sense inventories (WORDNET synsets and supersenses). Our final model is shown to perform comparably to state-of-the-art WSD systems.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

The rest of the paper is organized as follows: In Section 2, we describe current problems of WSD and related works. In Section 3, we describe background topics related to WordNet. In Section 4, we describe our model and the machine learning method that we use. In Section 5, 6, and 7, we present our experimental setup, the results, and an in-depth analysis on the contribution of the sense dependencies. Finally, in Section 8, we present our concluding remarks.

2 Problems and related works

2.1 Word sense dependencies

For the unsupervised[1] WSD, which aims to disambiguate polysemous words without using any tagged corpora, the use of sense dependencies has been a common method. (Mihalcea 2005) introduced an unsupervised graph-based algorithm, and shows the significant superiority of the sequence labeling model over the individual label assignment. (Sinha and Mihalcea 2007) built a model based on various word semantic similarity measures and graph centrality algorithms, which also used the graph structure that incorporates the sense dependencies. Thus, for unsupervised WSD, the effectiveness of sense dependencies has been shown by several researches, although the dependencies that they have considered are based only on information in WordNet, and are fixed in advance without any optimization for real texts.

On the contrary, most approaches to supervised WSD are to solve an independent classification problem for each word (e.g. (Decadt et al. 2004; Kohomban and Lee 2005; Tratz, Sanfilippo, Gregory, Chappell, Posse, and Whitney 2007)). These approaches have been developed along with research based on the lexical sample task (Mihalcea, Chklovski, and Kilgarriff 2004) in the Senseval s (Evaluation Exercises for the Semantic Analysis of Text). However, as described in Section 1, they cannot handle the interdependencies among word senses, and may output a semantically incoherent assignment of senses.

Recently, with the growing interest in the all-words task (Snyder and Palmer 2004), a few supervised WSD systems have incorporated the dependencies between word senses. SenseLearner (Mihalcea and Faruque 2004) and SuperSenseLearner (Mihalcea, Csomai, and Ciaramita 2007) incorporated sequential sense dependencies into the supervised WSD frameworks. (Ciaramita and Altun 2006) also took a sequential tagging approach for the disambiguation of WordNet supersenses. These approaches assume the sense dependencies between adjacent words, and

[1] There exists no common agreement on the definition of "unsupervised" WSD. More precisely, we mean "minimally-supervised" WSD, which requires only the sense inventory.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

optimize them based on tagged corpora. However, they cannot handle longer dependencies that are considered to be semantically dependent on each other (e.g. a verb and its object). Note additionally, that the dependencies that they have considered are between either WordNet synsets or supersenses, and hence are not combined with finer- or coarser-grained tag sets.

In this context, it is of interest to note whether the sense dependencies on a syntactic structure, rather than on a linear chain, work effectively or not. To the extent of our knowledge, there exists no WSD model that considers the interdependencies of word senses on a syntactic structure. Furthermore, despite the approaches described above, the contribution of sense dependencies for the supervised WSD has not been explicitly examined thus far. These questions are clarified by our research.

2.2 The use of coarse-grained tag sets

In Section 1, we presented one of the most significant issues in WSD—the data sparseness problem. This problem may even be magnified when we take into consideration the interdependencies of word senses, which are described as combinations of two word senses. In order to relieve this problem, we use the hierarchical information in the WordNet, including the superordinate words and supersenses, as described in Section 3. Although such information has never been combined with the sense dependencies, the use of the hierarchical information has been motivated by several different researches. For example, a WSD system by (Mihalcea and Faruque 2004), ranked second in the Senseval-3, consists of two models: the first model applied to words seen in the training data, and the second model that performs a generalized disambiguation process for words unseen in the data, by using the hierarchical information in the WordNet.

The fine granularity of the WordNet synsets is not just a major obstacle in achieving a high-performance WSD, but is sometimes *too fine-grained* even for a human to distinguish. This is reflected in the low inter-annotator agreement[2] of sense tagging, which implies that WSD models would be unlikely to perform better than this accuracy. On the other hand, (Ide and Veronis 1998) reported that coarse-grained sense distinctions are sufficient for several NLP applications. In particular, the use of the supersenses as the sense inventory[3] has recently been investigated by (Ciaramita and Altun 2006), and has received much attention in the WSD field. In this case, the inter-annotator agreements are turned out to reach nearly 90% (Navigli, Litkowski, and Hargraves 2007). For this reason, we use the WordNet supersenses, as well as the synsets, as the sense inventory for our experiments.

[2] typically around 65% (Mihalcea et al. 2004).
[3] the definition of word senses on which a WSD system is based.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

3 WordNet

The WordNet (Miller 1995) is a broad-coverage machine-readable dictionary (MRD) for English, containing about 150,000 words. WordNet also serves as an ontology, in which relations among words and word senses, and well-organized hierarchies of senses are defined. In this paper, we always refer to the WordNet version 2.0[4] unless otherwise noted. The statistics of the WordNet 2.0 is shown in Table 1 and 2.

In WordNet, nouns and verbs are organized into hierarchical structures with IS-A (hypernym–hyponym) relationships among words. All nouns and verbs, with the exception of some top-level concepts, are classified into primitive groups called *supersense*s, which we describe later. Figure 1 shows the WordNet hierarchical structure for the first sense (*financial bank*) of the noun *bank*, where each line indicates a synset with the list of synonymous words headed by its supersense label; an arrow denotes that the two synsets are in an IS-A relation. The synset {*group#1, grouping#1*} is a top-level broad semantic category that corresponds to[5] the supersense group *noun.group*. The lower synsets {*social group#1*}, {*organization#1, organisation#3*},

Table 1 WordNet 2.0 statistics

PoS	#word	#synset	#sense
Noun	114,648	79,689	141,690
Verb	11,306	13,508	24,632
Adjective	21,436	18,563	31,015
Adverb	4,669	3,664	5,808
Total	152,059	115,424	203,145

Table 2 WordNet 2.0 polysemy information (*mono.* denotes monosemous words)

PoS	Average Polysemy	
	incl. mono.	excl. mono.
Noun	1.23	2.79
Verb	2.17	3.66
Adjective	1.44	2.80
Adverb	1.24	2.49

```
<noun.group> depository financial institution#1, bank#1, banking concern#1, ...
   => <noun.group> financial institution#1, financial organization#1
      => <noun.group> institution#1, establishment#2
         => <noun.group> organization#1, organisation#3
            => <noun.group> social group#1
               => <noun.Tops> group#1, grouping#1
```

Fig. 1 WordNet hierarchical structure for a noun *bank#1*

[4] Note that the definition of synsets slightly differs according to the versions. Although this version is not up-to-date, we adopt this version because consistently-formatted SemCor and Senseval data sets for this are available.

[5] Note that since the classification of supersenses is not always consistent with the WordNet hierarchy, there are some cases in which a synset belongs to a different supersense than that of its parent's.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

and {*institution#1, establishment#2*} are more specific synsets, which in this paper we call the first, second, and third *general synset*s. Note that we use this hierarchical information for only nouns and verbs, because adjectives and adverbs do not have such hierarchical structures as they have.

3.1 Supersense

A *supersense* (Ciaramita, Hofmann, and Johnson 2003) is a coarse-grained semantic category, with which each noun or verb synset in WordNet is uniquely associated. Noun and verb synsets are associated with 26 and 15 categories, respectively. The coarse-grained sets of sense labels are easily distinguishable, and enable us to build a high-performance and robust model with small training data. We can also expect them to act as a good smoothing feature for WSD, which would make up for the sparseness of features associated with finer-grained senses. The effectiveness of using supersenses for WSD has recently been shown by several researchers (Kohomban and Lee 2005; Ciaramita and Altun 2006; Mihalcea et al. 2007). The complete lists of supersenses are shown below.

Noun supersense: act, animal, artifact, attribute, body, cognition, communication, event, feeling, food, group, location, motive, object, quantity, phenomenon, plant, possession, process, person, relation, shape, state, substance, time, Tops[6]

Verb supersense: body, change, cognition, communication, competition, consumption, contact, creation, emotion, perception, possession, social, stative, weather

3.2 Sense frequency information

Typically, senses of a word are not uniformly distributed. Since data sparsity has been a significant issue in WSD, sense frequency information is helpful in achieving a good performance. This information acts as a useful feature that offers a preference for frequent senses, and also as a back-off feature, which enables our model to output the first sense (or most frequent sense) when no other features are active for that word. We use the sense frequency information available in the WordNet, which is extracted from a standard, balanced corpus, the SemCor[7].

Due to the limitation of the computational time and memory, we incorporate the sense frequency information in a rather indirect manner, so that every feature can be described as a binary feature. For the supersense-based model, we use only the *first sense* (most frequent sense) of a

[6] *noun.Tops* is a special group in which several top-level synsets in the WordNet hierarchy are classified.
[7] Actually, the distribution of senses has turned out to vary according to the domain (McCarthy, Koeling, Weeds, and Carroll 2004). The utilization of the domain information is left as one of our future works.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

word as a feature. Since the first sense baseline is highly competitive[8] in the all-words WSD, this feature is expected to account for a substantial proportion of the sense frequency information. For the synset-based model, we alternatively use the *sense ranking* of a sense among its candidate senses. This is because the first sense feature is inappropriate when sufficient training instances are not available for every sense. Since senses of a word in WordNet are ordered according to frequency, it can represent the frequency of a sense in a simple way, while the sense distribution of every word is treated equally.

4 WSD model with tree-structured CRFs

4.1 Approach

In Section 2, we described two problems in the WSD field: the independent classification of each word's sense, and the scarcity of the training data. We address these problems by combining two methods.

The first method is the use of the syntactic dependencies of word senses on a dependency tree. In particular, we assume that there are strong dependencies of word senses between a syntactic head and its dependents in the dependency tree, rather than between neighboring words in the sentence. To the extent of our knowledge, our model is the first WSD model that incorporates the sense dependencies based on a syntactic structure.

The second method is the combination of various coarse-grained sense tag sets with the WordNet synsets. In our experiments, these tag sets are used in two ways. One way directly uses them as the sense inventory, instead of the finer sense inventory. In our *supersense-based model*, we use the supersenses as the sense inventory, and each word sense is disambiguated at the granularity level of supersenses. This method serves us much more training instances for each coarser *sense*, while we can no longer distinguish the finer senses inside it. The other way uses the coarse-grained tag sets in combination with finer sense tag sets. In our *synset-based model*, three coarse-grained label sets are incorporated in combination with the fine-grained WordNet synsets. Although the sense disambiguation is still based on the finer senses, the coarser sense tags will help the discrimination of the finer senses, by serving generalized information for each finer sense.

[8] In our experiment, our first sense classifier achieved the accuracies 65.3% for the Senseval-2 English all-words task test set, and 63.4% for the Senseval-3 English all-words task test set. Since the sense frequency information in WordNet is based on the SemCor, this baseline performs far better on the SemCor: 75.9% for the *brown1* section and 74.3% for the *brown2* section.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

Finally, the process of WSD is summarized below. At the beginning, we parse target sentences with a dependency parser, and compact the output trees so that they can describe informative dependencies among words, as described in Section 4.3. Then, the WSD task is regarded as a labeling task on the tree structures. By using T-CRFs, we can model this as the maximization of the probability of word sense trees, given the scores for vertices and edges. In the training phase, all vertex features and edge features are extracted using the gold-standard senses, and the weight vectors are optimized over the training data. In the testing phase, all possible combinations of senses are evaluated for each sentence, and the most probable sense assignment is selected.

4.2 Tree-structured conditional random fields

Conditional Random Fields (CRFs) are graph-based probabilistic discriminative models proposed by (Lafferty, McCallum, and Pereira 2001). CRFs are state-of-the-art methods for sequence labeling problems in many NLP tasks. CRFs construct a conditional model $p(\boldsymbol{y}|\boldsymbol{x})$ from a set of paired observations and label sequences. The conditional probability of a label sequence $\boldsymbol{y}$ conditioned on a data sequence $\boldsymbol{x}$ is given by

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left[\sum_{e\in E}\sum_{j}\lambda_j f_j(e,\boldsymbol{y}(e),\boldsymbol{x}) + \sum_{v\in V}\sum_{k}\mu_k g_k(v,\boldsymbol{y}(v),\boldsymbol{x})\right] \tag{1}$$

where f_j and g_k are the feature vectors for an edge and a vertex, λ_j and μ_k are the weight vectors, $\boldsymbol{y}_e$ and $\boldsymbol{y}_v$ are the set of components of $\boldsymbol{y}$ associated with an edge e and a vertex v, and $Z(x)$ is the partition function which constrains the sum of all the probabilities to be 1.

Tree-structured CRFs (T-CRFs) (Tang et al. 2006) are different from widely used linear-chain CRFs, in that the random variables are organized in a tree structure. Hence, they are appropriate for modeling the syntactic dependencies of word senses, which cannot be represented by linear structures. In this model, the optimal label assignment $\hat{\boldsymbol{y}}$ for an observation sequence $\boldsymbol{x}$ is calculated by

$$\begin{aligned}
\hat{\boldsymbol{y}} &= \operatorname*{argmax}_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}) \\
&= \operatorname*{argmax}_{\boldsymbol{y}} \frac{1}{Z(x)} \exp\sum_{v\in V}\left\{\sum_{j}\lambda_j f_j(v,\boldsymbol{x},\boldsymbol{y}(v)) + \sum_{k}\mu_k g_k(v,v',\boldsymbol{x},\boldsymbol{y}(v),\boldsymbol{y}(v'))\right\} \qquad (2)\\
&= \operatorname*{argmax}_{\boldsymbol{y}} \prod_{v\in V}\exp\left\{\sum_{j}\lambda_j f_j(v,\boldsymbol{x},\boldsymbol{y}(v)) + \sum_{k}\mu_k g_k(v,v',\boldsymbol{x},\boldsymbol{y}(v),\boldsymbol{y}(v'))\right\} \qquad (3)
\end{aligned}$$

where v denotes a vertex corresponding to a word while v' denotes the vertex corresponding to

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

its parent in the dependency tree. If we instead interpret v' as the vertex associated with the preceding word in a sentence, T-CRFs are reduced to linear-chain CRFs. Although T-CRFs are relatively new models, they have already been applied to several NLP tasks, such as semantic role labeling (Cohn and Blunsom 2005) and semantic annotation (Tang et al. 2006), proving to be useful in modeling the semantic structure of a text. Our model is the first application of T-CRFs to WSD.

4.3 Graph construction

In this section, we introduce the method of building graph structures on which CRFs are constructed. First, we describe how to construct a tree used in the tree-structured model. Let us consider the synset-level disambiguation of the following sentence.

(i) —*The man destroys confidence in banks.*

In the beginning, we parse this sentence with Sagae and Tsujii's dependency parser (Sagae and Tsujii 2007), which outputs parsed trees in the CoNLL-X dependency format (Buchholz and Marsi 2006). The left-hand side of Figure 2 shows the parsed tree for Sentence (i), where each child–parent edge denotes a directed dependency of words, and the labels on the edges denote the dependency types[9]. While this dependency tree describes dependencies among all words, including content words and function words, some of these dependencies are not informative for our WSD task, because our task does not focus on the disambiguation of function words. For example, on the left-hand side of Figure 2, the dependencies among *confidence*, *in*, and *bank* are split into the two dependencies *confidence–in* and *in–bank*; hence our model cannot capture the direct dependency between *confidence* and *bank*, which are considered to be semantically correlated. One way to solve this problem is to use higher-order (semi-Markov) dependencies, but this may drastically increase the computational cost. Thus, for the synset-based model, we convert the output dependency tree into a tree of content words, as exemplified on the right-hand side of Figure 2. In this process, the function words are removed from the tree, and their parent and child vertices are directly connected. The removed words[10] are included as a feature for the new edge. Now, the dependency between *confidence* and *bank* in Figure 2 is described as a direct edge. Thus, by the compaction of the trees, our model can capture more useful dependencies among word senses.

[9] For instance, ⟨SBJ⟩ denotes the subject–verb relation, and ⟨NMOD⟩ denotes the noun–modifier relation.
[10] If there are more than one function words between content words, the combination of these words are used (e.g. $in + that$). As the dependency label for the new edge, that of the uppermost edge is used.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

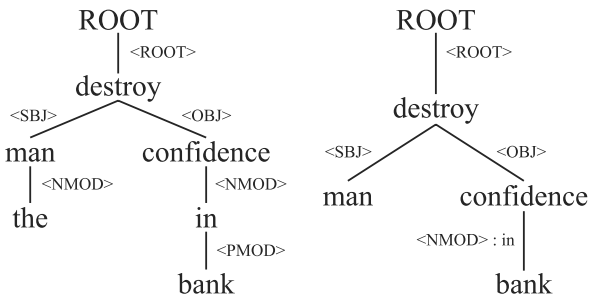

Fig. 2 An example sentence expressed as a dependency tree structure

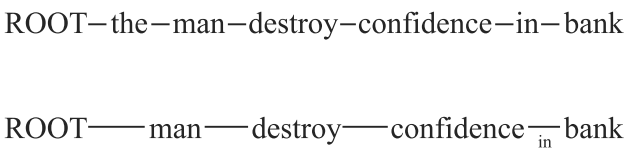

Fig. 3 An example sentence expressed as a linear chain structure

For the supersense-based model, we further convert the tree into a tree of nouns and verbs, because supersenses are defined for only these two parts of speech. The inclusion of removed words and dependency relation labels are performed in exactly the same manner as in the synset-based model. Consequently, the tree on the right hand side of Figure 2 remains unchanged, because in this case the sentence does not contain any adjectives nor adverbs.

For the linear-chain models, parsing is unnecessary. At first, we connect every adjacent words with an edge, and build a linear chain. Next, for the same reason as for the tree-structured case, we remove those words that we do not need to disambiguate from the graph, in order to capture the direct dependencies between content words (or nouns and verbs in the supersense-based model). The process of compacting the tree is described in Figure 3.

4.4 Example

In this section, let us present an intuitive illustration of how our model works. Here, we focus on three words *destroy*, *confidence*, and *bank* in Sentence (i). For simplicity, we consider only two major senses for each word as described in Table 3, so that the number of possible sense assignments is $2^3 = 8$. After an appropriate compaction of the dependency tree, dependencies among *destroy*, *confidence*, and *bank*, are represented as direct connections. Now, our objective is to determine the correct assignment of senses to these words, given the trained weight vector for features. We conduct this by evaluating the scores for all possible assignment of senses.

Let us start from the dependency between *confidence* and *bank*. The first intuition would be that *confidence(n)#2* is strongly related to a group or an institution (*financial bank*), but is unrelated to a natural landscape (*river bank*), while *confidence(n)#1* depends mostly on persons and not on other entities. Because *bank* does not have a "person" meaning, the weight of *confidence(n)#2–bank(n)#1* is expected to be higher than those of other possible sense bigrams.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

Table 3 Senses for *destroy*, *confidence*, and *bank*

destroy(v)#1	destruct, cause the destruction or undoing of
destroy(v)#2	ruin, damage irreparably
confidence(n)#1	self-confidence, belief in yourself
confidence(n)#2	a feeling of trust
bank(n)#1	depository financial institution
bank(n)#2	sloping land

A similar argument can be made for the dependency between *destroy* and *confidence*. We can assume that *destroy(v)#1* is usually associated with real objects, whereas *destroy(v)#2* can take either a real entity or an abstract thing as its direct object. Given *confidence* does not have an "object" meaning, the weights of *destroy(v)#2–confidence(n)#1* and *destroy(v)#2–confidence(n)#2* would be largest among others. Finally, given all scores for these sense dependencies, we can evaluate the overall score for the sentence, and see $\langle$*destroy(v)#2*, *confidence(n)#2*, *bank(n)#1*$\rangle$ is the most probable assignment of senses.

In practice, specific bigrams of synsets such as *confidence(n)#2–bank(n)#1* and *destroy(v)#2–confidence(n)#2* may not appear in the training data. In this case, sense bigrams combined with coarser sense labels work effectively. For example, if there are synset bigrams such as *destroy(v)#2–affection(n)#1* in the training data, the model can still perform the disambiguation process properly by considering a generalized synset–supersense bigram *destroy(v)#2–noun.feeling*. The detailed description of sense bigrams are provided in Section 4.7.

4.5 Sense labels

Using information in the WordNet, we make use of four sense labels for each word: a synset S_{WS}, two general synsets S_{G1} and S_{G2}, and a supersense S_{SS}, which we introduced in Section 3. These labels represent word senses at different granularity levels, and are to be combined with the vertex and edge features. We hereafter distinguish each sense label by putting one of the prefixes *WS*, *G1*, *G2*, and *SS*, as in *WS:bank#1* and *SS:noun.group*. The example of the sense labels for *destroy(v)#1* is shown in Table 4. For the words other than nouns and verbs, the supersense *N/A* is assigned.

In our model, we combine the synset and supersense labels[11] with the vertex features, and combine all four sense labels with the edge features. We denote the set of sense labels for vertex features by $\mathcal{S}_{\mathrm{VT}}$ $(= \{S_{\mathrm{WS}}, S_{\mathrm{SS}}\})$, and the one for edge features by $\mathcal{S}_{\mathrm{ED}}$ $(= \{S_{\mathrm{WS}}, S_{\mathrm{G2}}, S_{\mathrm{G1}}, S_{\mathrm{SS}}\})$.

[11] The use of these two sense labels were most effective in our preliminary experiments.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

Table 4 An example of sense labels for *bank(n)#1*

S_{WS}	Synset	depository financial institution#1, bank#1, banking concern#1, banking company#1
S_{G2}	Second general synset	organization#1, organisation#3
S_{G1}	First general synset	social group#1
S_{SS}	Supersense	noun.group

4.6 Vertex features

4.6.1 Synset-based model

We implement as vertex features a set of typical contextual features widely used in many supervised WSD models. Most of these features are those used by (Lee and Ng 2002), with the exception of the syntactic features.

In order to see the effectiveness of sense dependency features, we include as vertex features the word forms, lemmas, and parts of speech of both the parent and the child words in the dependency tree. These features provide the syntactic information of the parent and child words that are not semantically disambiguated. Therefore, if the sense bigram features work effectively over these features, it clearly shows that there exist instances that cannot be disambiguated without considering the interdependency of word senses. The list of vertex features also includes the information of both the preceding and following words, which in the linear-chain model plays the same role as the parent and child information in the tree-structured model.

Below is the list of contextual information used for the vertex features in the synset-based model. We refer to these features as $\mathcal{F}_{\mathrm{VT}}(v)$.

- Word form (WF): word form as it appears in a text.
- Global context (GC): bag-of-words within a 60-word window.
- Local PoS (LP): LP(-3), LP(-2), LP(-1), LP(0), LP(1), LP(2), and LP(3), where i in LP(i) denotes the relative position to the target word.
- Local context (LC): LC(-2), LC(-1), LC(0), LC(1), LC(2), LC$(-2,-1)$, LC$(-1,1)$, LC$(1,2)$, LC$(-3,-1)$, LC$(-2,1)$, LC$(-1,2)$, and LC$(1,3)$, where LC(i) denotes the word at the relative position i, and LC(i,j) the n-gram from the relative position i to j.
- Syntactic context (SC): word forms, lemmas, and parts of speech of the parent and child words.

Using this contextual information $\mathcal{F}_{\mathrm{VT}}(v)$ and the set of vertex labels $\mathcal{S}_{\mathrm{VT}}$, we construct a set of features on a vertex v by $\mathcal{S}_{\mathrm{VT}}(v) \otimes \mathcal{F}_{\mathrm{VT}}(v)$. Additionally, we include the sense ranking feature

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

(see Section 3.2 for detail), which is not combined with any sense label nor with any contextual information.

4.6.2 Supersense-based model

For the supersense-based model, we use vertex features based on (Ciaramita and Altun 2006), which includes some features from the named entity recognition literature, including the word shape features, along with the standard feature set for WSD. As the sense frequency information, we use the first sense feature. Unlike in the synset-based model, we do not incorporate the syntactic information of the parent and child words, since it has been reported not to improve the performance by (Ciaramita and Altun 2006).

4.7 Edge features

We design a set of edge features that represents the inter-word sense dependencies. For each edge, we define the sense bigram features $\mathcal{S}_{\mathrm{ED}}(v) \otimes \mathcal{S}_{\mathrm{ED}}(v')$. Moreover, in addition to these simple bigrams, we define two kinds of combined bigrams: the sense bigrams with the dependency relation labels[12] (e.g. *WS:confidence#2–(NMOD)–WS:bank#1*), and the sense bigrams with removed words in between (e.g. *WS:confidence#2–in–WS:bank#1*). Consequently, the number of features for each edge is $4^2 \cdot 3 = 48$.

5 Experimental setup

5.1 Data sets

In this section, we introduce corpora that we have used for the evaluation. SemCor (Miller, Leacock, Tengi, and Bunker 1993) is a corpus, in which all content words are annotated with the WordNet synsets, and consists of balanced 352 files from the Brown Corpus. It is divided into three parts: *brown1*, *brown2*, and *brownv* sections. In *brown1* and *brown2*, all content words (nouns, verbs, adjectives, and adverbs) are semantically annotated, while in *brownv* only verbs are annotated. Also, we use two data sets from the Senseval exercises: the Senseval-2 English all-words task (Palmer, Fellbaum, Cotton, and Dang 2001) test set, consisting of three articles from the Wall Street Journal, and the Senseval-3 English all-words task (Snyder and Palmer 2004) test set, consisting of two articles from the Wall Street Journal and a fiction excerpt from the Brown corpus.

[12] In order to compare the linear-chain models with the tree-structured models, we incorporate these dependency labels into the linear-chain models as well.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

As the data sets for evaluation, we use the *brown1* and *brown2* sections (denoted as SEM) of SemCor, and the Senseval-2 and -3 all-words task test sets (denoted as SE2 and SE3, respectively). We use the converted versions[13] annotated with WordNet 2.0 synsets. In these data sets, multi-word expressions are already segmented, while they are not in the original corpora. However, on the other hand, our model cannot output any answers to multi-word expressions that have no directly corresponding WordNet synsets, because we treat each expression as one unit in the process of WSD. For example, the multi-word expression *tear-filled* is treated as one instance. But it is not tagged with any WordNet synsets in the converted corpus, while in the original corpus it is tagged with two WordNet synsets for *tear* and *filled*. For this reason, we exclude such instances beforehand, and evaluate our models only on expressions that have corresponding synsets in the WordNet. The resulting statistics[14] of the data sets are shown in Table 5.

The evaluation of our model is performed by splitting these corpora into training, development, and test sets. At first, all files in SEM are sorted according to their file names and distributed into five data sets in order (denoted as SEM-A, SEM-B, SEM-C, SEM-D, and SEM-E), so that each set has almost the same distribution of domains[15]. Next, each of these five data sets is again divided into two halves: SEM-A1, SEM-A2, $\cdots$, SEM-E1, and SEM-E2, also according to the order of file names.

Our evaluation is based on a 5-fold cross validation scheme. In the training phase, four sets (e.g. SEM-A, SEM-B, SEM-C, and SEM-D) in the SEM are used for training. Next, for the evaluation on SemCor, one half of the rest (e.g. SEM-E1) is used for development and the other half (e.g. SEM-E2) is used for evaluation. For the evaluation on the Senseval data sets, all instances of the rest (e.g. SEM-E) are used for development and one of the Senseval data sets (SE2 or SE3) is used for evaluation. Finally, for the comparison with state-of-the-art models,

Table 5 Statistics of data sets

	# Tagged word	# Tagged noun and verb
SEM	189,667	135,123
SE2	2,259	1,567
SE3	1,978	1,617

[13] These data sets are available at http://www.cs.unt.edu/~rada/downloads.html
[14] These figures do not include words that our system cannot handle, including the instances tagged as "U" (corresponding to senses that do not exist in WordNet, 34 instances), and the multi-word expressions that have no directly corresponding WordNet synset (29 instances).
[15] Each document in the SemCor is named according to the domain that it belongs to.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

our model is trained on the whole set of SEM, and SE2 and SE3 are used for development and evaluation respectively

Our T-CRF model is trained by using Amis (Miyao and Tsujii 2002). During the development phase, we tune the Gaussian parameter σ for the L_2 regularization term.

5.2 Evaluation and models

As the evaluation measure, we use the standard recall measure, which is equivalent to the precision as we output answers to all instances. We evaluate our models based on the recalls averaged over the five trials of the cross validation.

The *synset-based evaluation* is performed based on the WordNet synsets. We evaluate the outputs of our system for all instances that are semantically tagged in the data sets. Each target word is either a noun, verb, adjective, or adverb.

For the *supersense-based evaluation*, we follow most of the experimental setup in (Ciaramita and Altun 2006). As they noted, in the WordNet, there is semantically inconsistent labeling of supersenses such that top level synsets are tagged as the supersense *noun.Tops* rather than the specific supersense they govern. For example, nouns such as *peach* and *plum* are tagged as *noun.plant* but their hypernym *plant* belongs to *noun.Tops*. For this reason, we adopted the modification of noun supersenses in the same way as (Ciaramita and Altun 2006), substituting *noun.Tops* labels with more specific supersense labels when possible, and left some general nouns with *noun.Tops*[16]. The evaluation is based on these modified labels. We ignore the adjective and adverb instances in the supersense-based evaluation.

Table 6 is the list of models that we use for the evaluation, where FS and SR correspond to the first sense and sense ranking features respectively, and *non-dependency* denotes models that

Table 6 The list of models for evaluation

		Synset-based	Supersense-based
Tree-structured	With FS/SR	Tree-WS-SR	Tree-SS-FS
	Without FS/SR	Tree-WS	Tree-SS
Linear-chain	With FS/SR	Linear-WS-SR	Linear-SS-FS
	Without FS/SR	Linear-WS	Linear-SS
Non-dependency	With FS/SR	NoDep-WS-SR	NoDep-SS-FS
	Without FS/SR	NoDep-WS	NoDep-SS
Baseline (WordNet first sense)		Baseline-WS	Baseline-SS

[16] Nouns which are left with *noun.Tops* are: entity, thing, anything, something, nothing, object, living thing, organism, benthos, heterotroph, life, and biont.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

do not incorporate the sense dependency features.

6 Result

6.1 Contribution of sense dependencies

In this section, we focus on the contribution of the sense dependencies. Table 7 shows the comparisons between the tree-structured models with sense dependencies (*dependency models*) and the models without sense dependencies (*non-dependency models*). Each figure displays the mean recall (equivalent to the precisions) averaged over the five trials, the "Diff." rows show the differences between the dependency models and the non-dependency models, and † and ‡ denote the statistical significance of $p < 0.05$ and $p < 0.01$ respectively. From Table 7, it is seen that with the sense frequency information, the tree-structured models (statistically) significantly outperformed the non-dependency models on all the data sets. These improvements seem inconsiderable in figures; however, considering that for instance the No-Dep-SS-FS model outperforms the Baseline-SS model by only 0.37% on SEM, the further improvement of 0.21% is substantial, because it indicates that our dependency model could handle 57% more instances over the first sense baseline[17]. Note that, without the sense frequency information, the synset-based tree-structured model (Tree-WS) performed worse than the non-dependency model (NoDep-WS) on all the data sets, whereas the supersense-based model (Tree-SS) exhibited the robustness regardless of the existence of the sense frequency information. These results suggest that for the synset-based model, in which most synsets do not have enough instances in the training data, the combination with sense frequency information is necessary in order to avoid the data sparseness problem.

Table 7 The contribution of sense dependency features (tree-structured models)

	SEM	SE2	SE3
Tree-SS-FS	83.60%	78.93%	80.24%
NoDep-SS-FS	83.39%	78.24%	79.89%
Diff.	+0.21%‡	+0.69%‡	+0.35%†
Tree-SS	79.15%	77.78%	78.93%
NoDep-SS	79.11%	77.09%	78.24%
Diff.	+0.04%	+0.69%†	+0.53%
Baseline-SS	83.02%	76.26%	78.48%

	SEM	SE2	SE3
Tree-WS-SR	77.46%	68.51%	66.32%
NoDep-WS-SR	77.16%	67.87%	66.02%
Diff.	+0.29%‡	+0.64%‡	+0.30%†
Tree-WS	73.19%	68.38%	65.50%
NoDep-WS	73.27%	68.48%	65.88%
Diff.	−0.09%	−0.10%	−0.38%
Baseline-WS	75.06%	65.38%	63.40%

[17] Similarly, this corresponds to 35% and 25% improvements over the baseline on SE2 and SE3 respectively.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

Similarly, Table 8 shows the comparisons between the linear-chain dependency models and the non-dependency models. In the supersense-based evaluation, although the differences are slightly smaller than in the tree-structured models, we confirmed that the sense dependencies with the sense frequency information work effectively, with the overall improvements of 0.20–0.30% for the three data sets. However, without the frequency information, no statistically significant improvement nor deterioration is observed. In the synset-based evaluation, the overall trend is almost same as in the tree-structured case. Nonetheless, by the incorporation of the sense dependencies, the improvements with the sense frequency information were even less, and the deteriorations without the frequency information were even more than in the tree-structured case. These results are consistent with the results in the following section, where the tree-structured models are shown to outperform the linear-chain models.

6.2 Tree-structured CRFs vs linear-chain CRFs

In this section, let us focus on the difference between the tree-structured models and the linear-chain models. In Table 9, although some of the differences are marginal, we can see that the tree-structured models outperformed the linear-chain models, by focusing on the statistically significant differences. These results suggest that the dependencies on the tree structures capture

Table 8 The contribution of sense dependency features (linear-chain models)

	SEM	SE2	SE3
Linear-SS-FS	83.69%	78.44%	80.19%
NoDep-SS-FS	83.39%	78.24%	79.89%
Diff.	$+0.29\%^{\ddagger}$	$+0.20\%$	$+0.30\%^{\dagger}$
Linear-SS	79.12%	77.21%	78.22%
NoDep-SS	79.11%	77.09%	78.24%
Diff.	$+0.01\%$	$+0.12\%$	-0.21%
Baseline-SS	83.02%	76.26%	78.48%

	SEM	SE2	SE3
Linear-WS-SR	77.38%	67.89%	66.24%
NoDep-WS-SR	77.16%	67.87%	66.02%
Diff.	$+0.21\%^{\ddagger}$	$+0.02\%$	$+0.22\%$
Linear-WS	72.87%	67.68%	65.82%
NoDep-WS	73.27%	68.48%	65.88%
Diff.	$-0.41\%^{\ddagger}$	$-0.81\%^{\ddagger}$	-0.05%
Baseline-WS	75.06%	65.38%	63.40%

Table 9 The comparison of tree-structured models with linear-chain models

	SEM	SE2	SE3
Tree-SS-FS	83.60%	78.93%	80.24%
Linear-SS-FS	83.69%	78.44%	80.19%
Diff.	-0.08%	$+0.48\%^{\ddagger}$	$+0.05\%$
Tree-SS	79.15%	77.78%	78.96%
Linear-SS	79.12%	77.21%	78.22%
Diff.	$+0.03\%$	$+0.58\%^{\dagger}$	$+0.74\%^{\dagger}$

	SEM	SE2	SE3
Tree-WS-SR	77.46%	68.51%	66.32%
Linear-WS-SR	77.38%	67.89%	66.24%
Diff.	$+0.08\%$	$+0.62\%^{\ddagger}$	$+0.08\%$
Tree-WS	73.19%	68.38%	65.50%
Linear-WS	72.87%	67.68%	65.82%
Diff.	$+0.32\%^{\ddagger}$	$+0.71\%^{\dagger}$	-0.32%

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

more important characteristics than those on the linear chains do.

6.3 Contribution of coarse-grained sense labels

Table 10 shows the contributions of the coarse-grained sense labels. Whereas Tree-WS-SR and Tree-WS use all four sense labels for the edge features ($\mathcal{S}_{\mathrm{ED}} = \{S_{\mathrm{WS}}, S_{\mathrm{G2}}, S_{\mathrm{G1}}, S_{\mathrm{SS}}\}$), Tree-WS-SR' and Tree-WS' only use the synset labels ($\mathcal{S}_{\mathrm{ED}} = \{S_{\mathrm{WS}}\}$) so that we can see the contribution of the coarse-grained labels. Although the improvements are not statistically significant, we can see that the coarse-grained labels consistently did improve the performance on all the data sets.

6.4 Comparison with state-of-the-art models

Table 11 shows the comparison of our model with the state-of-the-art WSD systems. Since the evaluation here is performed with the SENSEVAL official scorer[18], the figures are slightly different than on our evaluation scheme used in the other sections. Our best model Tree-WS-SR outperformed the two best systems in the SENSEVAL-3 (Gambl (Decadt et al. 2004) and SenseLearner (Mihalcea and Faruque 2004)), but lagged behind PNNL (Tratz et al. 2007) by 1.6%. However, our model cannot handle multi-word expressions that do not exist in the WORDNET[19] as noted in Section 5.1, and all systems in Table 11 except for Simil-Prime (Kohomban and Lee

Table 10 The contribution of coarse-grained sense labels (tree-structured, synset-based models)

	SEM	SE2	SE3
Tree-WS-SR	77.46%	68.51%	66.32%
Tree-WS-SR'	77.40%	68.39%	66.06%
Diff.	+0.04%	+0.11%	+0.26%
NoDep-WS-SR	77.16%	67.87%	66.02%

	SEM	SE2	SE3
Tree-WS	73.19%	68.38%	65.50%
Tree-WS'	73.08%	68.21%	65.27%
Diff.	+0.11%	+0.17%	+0.23%
NoDep-WS	73.27%	68.48%	65.88%

Table 11 The comparison of the performance of WSD systems evaluated on the SENSEVAL-3 English all-words test set

System	Recall
PNNL (Tratz et al. 2007)	67.0%
Simil-Prime (Kohomban et al. 2005)	66.1%
Tree-WS-SR	*65.4%*
Gambl (Decadt et al. 2004)	65.2%
SenseLearner (Mihalcea et al. 2004)	64.6%
Baseline-WS	*62.2%*

[18] http://www.senseval.org/senseval3/scoring
[19] Our system cannot output any answers for these 29 instances, which correspond to the 1.5% recall.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

2005)[20] utilize other sense-annotated corpora, such as the SENSEVAL lexical sample task data sets or example sentences in the WORDNET, in addition to SEMCOR. Taking into consideration these factors, we can conclude that the performance of our T-CRF model is comparable to that of state-of-the-art WSD systems.

7 Discussion and analysis

7.1 Edge feature overview

Table 12 shows the list of the 15 largest-weighted sense dependency features in the tree-structured, synset-based model (Tree-WS). The list includes many features associated with adjective–noun relations (e.g. *SS:noun.person–WS:distinguished(a)#1*) and verb–noun relations (e.g. *WS:have(v)#2–SS:noun.attribute*). Hereinafter, λ denotes λ in Equation 3, and α denotes the exponential of λ. We call a feature either with a positive lambda or with an alpha larger than 1 as an *excitatory* feature, and that feature either with a negative lambda or an alpha smaller than 1 as an *inhibitory* feature.

Also, Table 13 shows the 15 largest-weighted sense dependency features in the linear-chain,

Table 12 Largest-weighted sense dependency features (tree-structured, synset-based model)

Feature	α	λ
SS:noun.person–WS:distinguished(a)#1	1.825031	0.6015970
SS:noun.person–ϵ–WS:distinguished(a)#1	1.825031	0.6015970
SS:noun.person–(NMOD)–WS:distinguished(a)#1	1.825031	0.6015970
SS:noun.person–WS:little(a)#4	1.718862	0.5416624
SS:noun.person–ϵ–WS:little(a)#4	1.718862	0.5416624
SS:noun.person–(NMOD)–WS:little(a)#4	1.718862	0.5416624
SS:noun.time–WS:last(a)#3#1	1.645755	0.4981992
SS:noun.time–ϵ–WS:last(a)#1	1.645755	0.4981992
SS:noun.time–(NMOD)–WS:last(a)#1	1.645755	0.4981992
SS:noun.person–ϵ–WS:old(a)#1	1.640665	0.4951016
SS:noun.person–WS:old(a)#1	1.640664	0.4951010
SS:noun.person–(NMOD)–WS:old(a)#1	1.640664	0.4951010
WS:have(v)#2–ϵ–SS:noun.attribute	1.639619	0.4944639
WS:have(v)#2–(OBJ)–SS:noun.attribute	1.637224	0.4930021
WS:have(v)#2–SS:noun.attribute	1.613662	0.4785061

[20] (Kohomban and Lee 2005) used almost the same training data as our system, but they utilize the instance weighting technique and the combination of several classifiers, which our system does not.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

Table 13 Largest-weighted sense dependency features (linear-chain, synset-based model)

Feature	α	λ
SS:noun.artifact–SS:noun.artifact	1.956750	0.6712849
SS:noun.communication–SS:noun.communication	1.897139	0.6403470
WS:distinguished(a)#1–ϵ–SS:noun.person	1.869355	0.6255935
WS:distinguished(a)#1–SS:noun.person	1.823676	0.6008542
WS:be(v)#1–ϵ–SS:N/A	1.816105	0.5966941
G1:be(v)#1–(AMOD)–SS:N/A	1.737981	0.5527241
WS:so(r)#2–SS:N/A	1.717166	0.5406753
SS:noun.substance–SS:noun.substance	1.686647	0.5227425
SS:noun.person–SS:noun.person	1.674598	0.5155731
SS:verb.motion–SS:noun.location	1.662030	0.5080397
WS:clear(a)#1–SS:N/A	1.646610	0.4987186
WS:be(v)#1–(AMOD)–SS:N/A	1.638383	0.4937098
WS:little(a)#4–ϵ–SS:noun.person	1.604844	0.4730266
WS:little(a)#4–SS:noun.person	1.591441	0.4646399
WS:well(r)#1–SS:N/A	1.577307	0.4557190

synset-based model. When compared to the outputs of the tree-structured model, we can see that the linear-chain model captures more successive noun–noun dependencies, while the tree-structured model captures more adjective–noun and verb–object dependencies. Thus, although the difference of the recalls is small, we can assume that the sense dependency features in the tree-structured model and those in the linear-chain model have different contributions to the results. The simultaneous use of both is of interest; however, since it makes our model no longer a tree, the implementation is not straightforward. Hence, this is left as one of our future works.

7.2 Instance-based analysis

7.2.1 Overview

In this section, we present instance-based analyses based on the first 100 instances for which the answer of the dependency model Tree-WS-SR differs from that of the non-dependency model NoDep-WS-SR in the first trial on SemCor. We extracted only the largest-weighted edge feature for each instance, assuming that this feature had the largest contribution to the result. These instances consist of 54 *positive instances*, for which Tree-WS-SR output the correct answer while NoDep-WS-SR did not, and 46 *negative instances*, for which Tree-WS-SR did not output the correct answer while NoDep-WS-SR did. Table 14 and 15 show the count of each edge type for these instances. For both positive and negative instances, the verb–noun dependencies are the dominant dependencies, which account for 48% of all the instances. One noteworthy point

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

Table 14 The counts of the largest-weighted dependency types in positive instances

Type	Count	Ratio
verb–noun	26	48.1%
noun–noun	10	18.5%
noun–adj	4	7.4%
verb–verb	3	5.6%
noun–mod	3	5.6%
verb–mod	3	5.6%
etc.	5	9.3%

Table 15 The counts of the largest-weighted dependency types in negative instances

Type	Count	Ratio
verb–noun	22	47.8%
verb–verb	6	13.0%
noun–noun	5	10.9%
noun–mod	4	8.7%
verb–adv	3	6.5%
verb–mod	2	4.3%
etc.	4	8.7%

is that more noun–noun dependencies are found in the positive instances than in the negative instances, which might suggest that noun–noun dependencies are particularly likely to capture useful dependencies and contribute to positive instances.

7.2.2 Verb–noun dependencies

Let us present two instances in which the verb–noun dependencies worked effectively. The first sentence is:

From this earth, then, while it was still virgin God took dust and fashioned the man, the beginning of humanity.

The verb *take* has as many as 42 senses in the WordNet. But fortunatelly, the first six senses belong to different supersenses, and our dependency model succeeded in outputting the correct sense *take#4* (*SS:verb.contact*, *take physically*) by making use of the strong dependency *SS:verb.contact–SS:noun.substance* ($\alpha = 1.209, \lambda = 0.1898$), given *dust#1* belongs to *noun.substance*.

The second instance is also a positive instance from the SEM-A data set.

For a serious young man who plays golf with a serious intensity, Palmer has such an inherent sense of humor that it relieves the strain and keeps his nerves from jangling like banjo strings.

Here, *has* is an ambiguous verb that has 19 senses in the WordNet. The correct sense here is *have(v)#2* (*SS:verb.stative*, *have as a feature*). Given *sense-of-humor#1* belongs to the supersense *noun.attribute*, the correct sense was output by the strong verb–object dependency *G1:have(v)#2–(OBJ)–SS:noun.attribute* ($\alpha = 1.331, \lambda = 0.2860$). While this verb–object dependency had a large excitatory weight, the corresponding verb–subject dependency had an inhibitory weight (*G1:have(v)#2–(SBJ)–SS:noun.attribute* ($\alpha = 0.919, \lambda = -0.0845$)), which in-

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

dicates that the dependency relation label also contributed to the result. Note also that this long-distance dependency cannot be described by linear-chain models.

Next, let us show a typical negative example, where a verb–subject dependency worked inappropriately.

> *The repeated efforts in Christian history to describe death as altogether the consequence of human sin show that these two aspects of death cannot be separated.*

The correct sense for *show* here is *show#2* (*verb.cognition*, *establish the validity*), but the model output *show#3* (*verb.communication*, *prove evidence for*) affected by the long dependency *WS:testify(v)#2–(SBJ)–SS:noun.act* ($\alpha = 1.186, \lambda = 0.1706$) between *efforts* and *show*. This subject-verb information seems to be inadequate for the disambiguation of *show*.

7.2.3 Noun–noun dependencies

Next, we focus on the noun–noun dependencies. The first example is a negative instance.

> *Philadelphia permitted him to seek a better connection after he had refused to reconsider his decision to end his career as a player.*

The noun *career* has two meanings: *the particular occupation for which you are trained* (*career#1*) and *the general progression of your working or professional life* (*career#2*). From the phrase *career as a player*, we might first assume that the correct sense of *career* can be either of two senses, with the possibility that there is a preference for *career#2*, just as explained by the largest-weighted dependency *WS:career(n)#2–(NMOD)–SS:noun.person* ($\alpha = 1.071, \lambda = 0.06859$) between *career* and *player*. However, in fact, the correct answer here is *career#1*, and the determining clue for this instance seems to be the verb–object dependency between *end* and *career*, which was not captured by our model.

Among the ten positive instances of the noun–noun dependencies, four instances were contributed by the *noun–of–noun* dependencies. Since dependencies of this type were not observed in the negative instances, they seem to particularly contribute to the positive instances. Let us consider the following example.

> *The embarrassment of these theories over the naturalness of death is an illustration of the thesis that death cannot be only a punishment, for some termination seems necessary in a life that is lived within the natural order of time and change.*

Although the correct sense *time#5* (*noun.Tops*, *the continuum of experience in which events pass from the future through the present to the past*) is not a frequent sense, our model correctly output this sense by using the dependency *SS:noun.object–of–WS:time(n)#5* ($\alpha = 1.054, \lambda = 0.05259$), given *natural order#1* belongs to the supersense *noun.object*.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

7.2.4 Coordination dependencies

Through our analysis, we observed that the noun–noun dependencies in coordination relations work remarkably well. In the following sentence, three words *nails*, *levels*, and *T squares* are in a coordination relation.

> *He also bought a huge square of pegboard for hanging up his tools, and lumber for his workbench, sandpaper and glue and assorted nails, levels and T squares and plumb lines and several gadgets that he had no idea how to use or what they were for.*

Here, the correct sense for *nail* is *nail#2* (*noun.artifact*, *a thin pointed piece of metal*) and that for *level* is *level#5* (*noun.artifact*, *indicator of the horizontal*). The relatively low frequency of these senses prevents our model from outputting the correct senses in an ordinal way. However, the dependency model could capture the fact that two words in a coordination relation are quite likely to belong to the same semantic group (*SS:noun.artifact–(COORD)–SS:noun.artifact* ($\alpha = 1.367, \lambda = 0.3126$)), and hence succeeded in the correct disambiguation of all these three words. More generally, we have observed that the coordination features for an edge that connects the same supersense all have positive weights.

8 Conclusion

In this paper, we proposed a novel approach for the all-words WSD, focusing on the use of syntactic dependencies of word senses, and investigated the contribution of these dependencies to WSD. Our proposals were twofold: to consider the sense dependencies on dependency trees, and to use the combined bigrams of fine- and coarse-grained senses as edge features.

In our experiments, the sense dependency features were shown to work effectively for WSD, with a 0.29%, 0.64%, and 0.30% improvement of recalls for SemCor, Senseval-2, and Senseval-3 data sets respectively. Despite the small improvements in overall figures, these improvements indeed correspond to 11–26% improvements over the first sense baseline. The dependency tree structures were shown to be appropriate in modeling the dependencies of word senses, by the results that the tree-structured models outperformed the linear-chain models. In the analysis section, we presented an in-depth analysis of the outputs, and observed that the noun–noun dependencies particularly contribute to the positive instances.

In addition, the combination of coarse-grained tag sets with the sense dependency features consistently improved the performance of WSD on every data set, although the improvements were not statistically significant. However, our experiments also showed that even when combined with the coarse-grained tag sets, the sense dependency features do not improve the performance

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

of WSD unless combined with proper sense frequency information relieving the data sparseness problem. The supersense-based WSD models, on the contrary, exhibited the robustness regardless of the existence of the sense frequency information.

The performance of our tree-structured model was comparable to that of the state-of-the-art WSD systems. Although our model was based on a simple framework and was trained only on the SemCor corpus, the results that we gained were promising, suggesting that our model still has a great potential for improvement. Our next interest is to combine our framework with the recently-developed semi-supervised frameworks. The combination of the local and syntactic dependencies with the global information is expected to further the WSD research.

Acknowledgment

This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan).

Reference

Boyd-Graber, J., Blei, D., and Zhu, X. (2007). "A Topic Model for Word Sense Disambiguation." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Buchholz, S. and Marsi, E. (2006). "CoNLL-X shared task on multilingual dependency parsing." In *Proceedings of CoNLL 2006*.

Ciaramita, M. and Altun, Y. (2006). "Broad-Coverage Sense Disambiguation and Information Extraction with a Supersense Sequence Tagger." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ciaramita, M., Hofmann, T., and Johnson, M. (2003). "Hierarchical Semantic Classification: Word Sense Disambiguation with World Knowledge." In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*.

Cohn, T. and Blunsom, P. (2005). "Semantic Role Labelling with Tree Conditional Random Fields." In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)*.

Decadt, B., Hoste, V., Daelemans, W., and den Bosch, A. V. (2004). "GAMBL, genetic algorithm optimization of memory-based WSD." In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

Gliozzo, A., Giuliano, C., and Strapparava, C. (2005). "Domain Kernels for Word Sense Disambiguation." In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*.

Hoste, V., Hendrickx, I., Daelemans, W., and van den Bosch, A. (2002). "Parameter optimization for machine-learning of word sense disambiguation." *Natural Language Engineering*, **8**, pp. 311–328.

Ide, N. and Veronis, J. (1998). "Word sense disambiguation: The state of the art." *Computational Linguistics*, **24**, pp. 1–40.

Kohomban, U. S. and Lee, W. S. (2005). "Learning Semantic Classes for Word Sense Disambiguation." In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*.

Lafferty, J., McCallum, A., and Pereira, F. (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data." In *Proceedings of 18th International Conference on Machine Learning (ICML)*.

Lee, Y. K. and Ng, H. T. (2002). "An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2004). "Finding predominant senses in untagged text." In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mihalcea, R. and Faruque, E. (2004). "SenseLearner: Minimally supervised word sense disambiguation for all words in open text." In *Proceedings of ACL/SIGLEX Senseval-3*.

Mihalcea, R. (2004). "Co-training and self-training for word sense disambiguation." In *Proceedings of CoNLL 2004*.

Mihalcea, R. (2005). "Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling." In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

Mihalcea, R., Chklovski, T., and Kilgarriff, A. (2004). "The Senseval-3 English lexical sample task." In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

Mihalcea, R., Csomai, A., and Ciaramita, M. (2007). "UNT-Yahoo: SuperSenseLearner: Combining SenseLearner with SuperSense and other Coarse Semantic Features." In *Proceedings of the 4th International Workshop on the Semantic Evaluations (SemEval-2007)*.

Miller, G. (1995). "Wordnet: A lexical database." *Communication of the ACM*, **38** (11),

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

pp. 39–41.

Miller, G. A., Leacock, C., Tengi, R., and Bunker, R. T. (1993). "A semantic concordance." In *Proceedings of the workshop on Human Language Technology.*

Miyao, Y. and Tsujii, J. (2002). "Maximum entropy estimation for feature forests." In *Proceedings of Human Language Technology Conference (HLT 2002).*

Navigli, R., Litkowski, K., and Hargraves, O. (2007). "Semeval-2007 task 07: Coarse-grained english all-words task." In *Proceedings of the Workshop on Semantic Evaluations (SemEval).*

Palmer, M., Fellbaum, C., Cotton, S., and Dang, H. T. (2001). "English Tasks: All Words and Verb Lexical Sample." In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2).*

Sagae, K. and Tsujii, J. (2007). "Dependency parsing and domain adaptation with LR models and parser ensembles." In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL.*

Sinha, R. and Mihalcea, R. (2007). "Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity." In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007)* Irvine, CA.

Snyder, B. and Palmer, M. (2004). "The English all-words task." In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text.*

Tang, J., Hong, M., Li, J., and Liang, B. (2006). "Tree-structured Conditional Random Fields for Semantic Annotation." In *Proceedings of the 5th International Semantic Web Conference.*

Tratz, S., Sanfilippo, A., Gregory, M., Chappell, A., Posse, C., and Whitney, P. (2007). "PNNL: A Supervised Maximum Entropy Approach to Word Sense Disambiguation." In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval).*

Jun Hatori: graduated from the Department of Earth and Planetary Physics, University of Tokyo, in 2007. After receiving his master's degree from the University of Tokyo in 2009, he has been in the doctoral course in the Graduate School of Information Science and Technology, University of Tokyo.

Yusuke Miyao: received the BSc and MSc from the University of Tokyo in 1998 and in 2000 respectively, and the PhD from the University of Tokyo in 2006. He is a research associate at the University of Tokyo from 2001. Member of Information Processing Society of Japan, Association for Computational Linguistics.

Information and Media Technologies 4(4) : 1129–1155 (2009)
reprinted from: Journal of Natural Language Processing 16(5) : 51–77 (2009)

Jun'ichi Tsujii: received the BE, ME, and PhD degrees from Kyoto University, Japan, in 1971, 1973, and 1978, respectively. He is currently a professor in the University of Tokyo (1995–) as well as a professor in the University of Manchester (1988–1995, 2004–). He is also Research Director of National Centre for Text Mining (NaCTeM), UK. He received Achievement Award of JSAI (2008). Member of ACL, President of ACL (2006), Permanent member of ICCL (from 1992).

(Received March 16, 2009)
(Revised May 12, 2009)
(Accepted August 19, 2009)