Word Sense Disambiguation
by Relative Selection

Hee-Cheol Seo[1], Hae-Chang Rim[2], and Myung-Gil Jang[1]

[1] Knowledge Mining Research Team,
Electronics and Telecommunications Research Institute (ETRI),
Daejeon, Korea
{hcseo, mgjang}@etri.re.kr
[2] Dept. of Computer Science and Engineering, Korea University,
1, 5-ka, Anam-dong, Seongbuk-Gu, Seoul, 136-701, Korea
rim@nlp.korea.ac.kr

Abstract. This paper describes a novel method for a word sense disambiguation that utilizes relatives (i.e. synonyms, hypernyms, meronyms, etc in WordNet) of a target word and raw corpora. The method disambiguates senses of a target word by selecting a relative that most probably occurs in a new sentence including the target word. Only one co-occurrence frequency matrix is utilized to efficiently disambiguate senses of many target words. Experiments on several English datum present that our proposed method achieves a good performance.

1 Introduction

With its importance, a word sense disambiguation (WSD) has been known as a very important field of a natural language processing (NLP) and has been studied steadily since the advent of NLP in the 1950s. In spite of the long study, few WSD systems are used for practical NLP applications unlike part-of-speech (POS) taggers and syntactic parsers. The reason is because most of WSD studies have focused on only a small number of ambiguous words based on sense tagged corpus. In other words, the previous WSD systems disambiguate senses of just a few words, and hence are not helpful for other NLP applications because of its low coverage.

Why have the studies about WSD stayed on the small number of ambiguous words? The answer is on sense tagged corpus where a few words are assigned to correct senses. Since the construction of the sense tagged corpus needs a great amount of times and cost, most of current sense tagged corpora contain a small number of words less than 100 and the corresponding senses to the words. The corpora, which have sense information of all words, have been built recently, but are not large enough to provide sufficient disambiguation information of the all words. Therefore, the methods based on the sense tagged corpora have difficulties in disambiguating senses of all words.

In this paper, we proposed a novel WSD method that requires no sense tagged corpus[1] and that identifies senses of all words in sentences or documents, not a small number of words. Our proposed method depends on raw corpus, which is relatively very large, and on WordNet [1], which is a lexical database in a hierarchical structure.

2 Related Works

There are several works for WSD that do not depend on a sense tagged corpus, and they can be classified into three approaches according to main resources used: raw corpus based approach [2], dictionary based approach [3,4] and hierarchical lexical database approach. The hierarchical lexical database approach can be reclassified into three groups according to usages of the database: gloss based method [5], conceptual density based method [6,7] and relative based method [8,9,10]. Since our method is a kind of the relative based method, this section describes the related works of the relative based method.

[8] introduced the relative based method using International Roget's Thesaurus as a hierarchical lexical database. His method is conducted as follows: 1) Get relatives of each sense of a target word from the Roget's Thesaurus. 2) Collect example sentences of the relatives, which are representative of each sense. 3) Identify salient words in the collective context and determine weights for each word. 4) Use the resulting weights to predict the appropriate sense for the target word occurring in a novel text. He evaluated the method on 12 English nouns, and showed over than 90% precision. However, the evaluation was conducted on just a small part of senses of the words, not on all senses of them.

He indicated that a drawback of his method is on the ambiguous relative: just one sense of the ambiguous relative is usually related to a target word but the other senses of the ambiguous relatives are not. Hence, a collection of example sentences of the ambiguous relative includes the example sentences irrelevant to the target word, which prevent WSD systems from collecting correct WSD information. For example, an ambiguous word *rail* is a relative of a meaning *bird* of a target word *crane* at WordNet, but the word *rail* means *railway* for the most part, not the meaning related to *bird*. Therefore, most of the example sentences of *rail* are not helpful for WSD of *crane*. His method has another problem in disambiguating senses of a large number of target words because it requires a great amount of time and storage space to collect example sentences of relatives of the target words.

[9] followed the method of [8], but tried to resolve the ambiguous relative problem by using just unambiguous relatives. That is, the ambiguous relative *rail* is not utilized to build a training data of the word *crane* because the word *rail* is ambiguous. Another difference from [8] is on a lexical database: they utilized WordNet as a lexical database for acquiring relatives of target words

---

[1] Strictly speaking, our method utilizes bias of word senses at WordNet, which is acquired a sense tagged corpus. However, our method does not access a sense tagged corpus directly. Hence, our method is a kind of a weakly supervised approach.

instead of International Roget's Thesaurus. Since WordNet is freely available for research, various kinds of WSD studies based on WordNet can be compared with the method of [9]. They evaluated their method on 14 ambiguous nouns and achieved a good performance comparable to the methods based on the sense tagged corpus. However, the evaluation was conducted on a small part of senses of the target words like [8].

However, many senses in WordNet do not have unambiguous relatives through relationships such as synonyms, direct hypernyms, and direct hyponyms.[2] A possible alternative is to use the unambiguous relatives in the long distance from a target word, but the way is still problematic because the longer the distance of two senses is, the weaker the relationship between them is. In other words, the unambiguous relatives in the long distance may provide irrelevant examples for WSD like ambiguous relatives. Hence, the method has difficulties in disambiguating senses of words that do not have unambiguous relatives near the target words in the WordNet. The problem becomes more serious when verbs, which most of the relatives are ambiguous, are disambiguated. Like [8], the method also has a difficulty in disambiguating senses of many words because the method collects the example sentences of relatives of many words.

[10] reimplemented the method of [9] using a web, which may be a very large corpus, in order to collect example sentences. They built training datum of all noun words in WordNet whose size is larger than 7GB, but evaluated their method on a small number of nouns of lexical sample task of SENSEVAL-2 as [8] and [9].

3   Word Sense Disambiguation by Relative Selection

Our method disambiguates senses of a target word in a sentence by selecting only a relative among the relatives of the target word that most probably occurs in the sentence. A flowchart of our method is presented in Figure 1 with an example[3]: 1) Given a new sentence including a target word, a set of relatives of the target word is created by looking up in WordNet. 2) Next, the relative that most probably occurs in the sentence is chosen from the set. In this step, co-occurrence frequencies between relatives and words in the sentence are used in order to calculate the probabilities of relatives. Our method does not depend on the training data, but on co-occurrence frequency matrix. Hence in our method, it is not necessary to build the training data, which requires too much time and space. 3) Finally, a sense of the target word is determined as the sense that is related to the selected relative. In this example, the relative *stork* is selected with the highest probability and the proper sense is determined as *crane#1*, which is related to the selected relative *stork*.

---

[2] In this paper, direct hypernyms and direct hyponyms mean parents and children at a lexical database, respectively.

[3] In WordNet 1.7.1, a word *crane* contains four senses, but in this paper only two senses (i.e. *bird* and *device*) are described in the convenience of description.
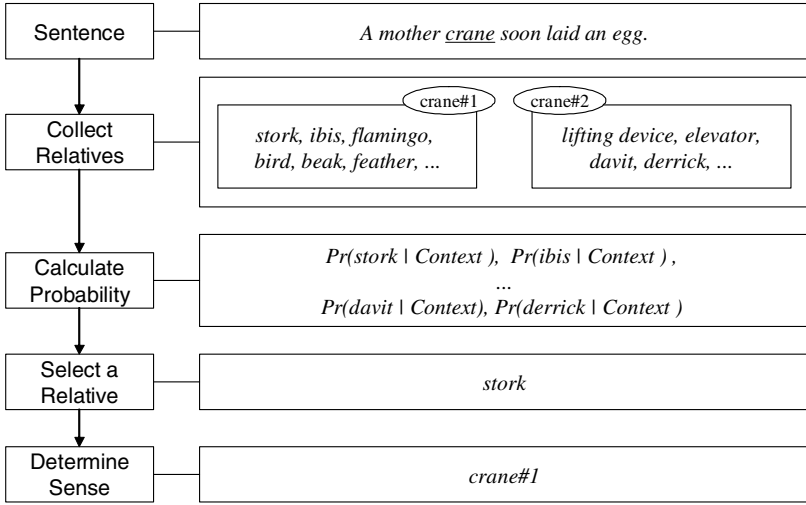
| Sentence | *A mother <u>crane</u> soon laid an egg.* |

crane#1 — *stork, ibis, flamingo, bird, beak, feather, ...*

crane#2 — *lifting device, elevator, davit, derrick, ...*

| Collect Relatives | |

| Calculate Probability | *Pr(stork | Context), Pr(ibis | Context),*
...
Pr(davit | Context), Pr(derrick | Context) |

| Select a Relative | *stork* |

| Determine Sense | *crane#1* |

Fig. 1. Flowchart of our proposed method

Our method makes use of ambiguous relatives as well as unambiguous relatives unlike [9] and hence overcomes the shortage problem of relatives and also reduces the problem of ambiguous relatives in [8] by handling relatives separately instead of putting example sentences of the relatives together into a pool.

3.1 Relative Selection

The selected relative of the i-th target word tw_i in a sentence C is defined to be the relative of tw_i that has the largest co-occurrence probability with the words in the sentence:

$$SR(tw_i, C)$$
$$\stackrel{\text{def}}{=} \arg\max_{r_{ij}} P(r_{ij}|C)P(S_{r_{ij}})W(r_{ij}, tw_i) \qquad (1)$$

where SR is the selected relative, r_{ij} is the j-th relative of tw_i, $S_{r_{ij}}$ is a sense of tw_i that is related to the relative r_{ij}, and W is a weight of r_{ij}. The right hand side of Eq. 1 is logarithmically calculated by Bayesian rule:

$$\arg\max_{r_{ij}} P(r_{ij}|C)P(S_{r_{ij}})W(r_{ij}, tw_i)$$
$$= \arg\max_{r_{ij}} \frac{P(C|r_{ij})P(r_{ij})}{P(C)}P(S_{r_{ij}})W(r_{ij}, tw_i)$$
$$= \arg\max_{r_{ij}} P(C|r_{ij})P(r_{ij})P(S_{r_{ij}})W(r_{ij}, tw_i)$$
$$= \arg\max_{r_{ij}} \{logP(C|r_{ij}) + logP(r_{ij})$$
$$+ logP(S_{r_{ij}}) + logW(r_{ij}, tw_i)\} \qquad (2)$$

The first probability in Eq. 2 is computed under the assumption that words in C occur independently as follows:

$$logP(C|r_{ij}) \approx \sum_{k=1}^{n} logP(w_k|r_{ij}) \tag{3}$$

where w_k is the k-th word in C and n is the number of words in C. The probability of w_k given r_{ij} is calculated:

$$P(w_k|r_{ij}) = \frac{P(r_{ij}, w_k)}{P(r_{ij})} \tag{4}$$

where $P(r_{ij}, w_k)$ is a joint probability of r_{ij} and w_k, and $P(r_{ij})$ is a probability of r_{ij}.

Other probabilities in Eq. 2 and 4 are computed as follows:

$$P(r_{ij}, w_k) = \frac{freq(r_{ij}, w_k)}{CS} \tag{5}$$

$$P(r_{ij}) = \frac{freq(r_{ij})}{CS} \tag{6}$$

$$Pr(S_{r_{ij}}) = \frac{0.5 + WNf(S_{r_{ij}})}{n * 0.5 + WNf(tw_i)} \tag{7}$$

where $freq(r_{ij}, w_k)$ is the frequency that r_{ij} and w_k co-occur in a raw corpus, $freq(r_{ij})$ is the frequency of r_{ij} in the corpus, and CS is a corpus size, which is the sum of frequencies of all words in the raw corpus. $WNf(S_{r_{ij}})$ and $WNf(tw_i)$ is the frequency of a sense related to r_{ij} and tw_i in WordNet.[4] In Eq. 7, 0.5 is a smoothing factor and n is the number of senses of tw_i. Finally, in Eq. 2, the weights of relatives, $W(r_{ij}, tw_i)$, are described in following Section 3.1.

Relative Weight. WordNet provides relatives of words, but all of them are not useful for WSD. That is to say, it is clear that most of ambiguous relatives may bring about a problem by providing example sentences irrelevant to the target word to WSD system as described in the previous section.

However, WordNet as a lexical database is classified as a fine-grained dictionary, and consequently some words are classified into ambiguous words though the words represent just one sense in the most occurrences. Such ambiguous relatives may be useful for WSD of target words that are related to the most frequent senses of the ambiguous relatives. For example, a relative *bird* of a word *crane* is an ambiguous word, but it usually represents one meaning, *"warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings"*,

---

[4] WordNet provides the frequencies of words and senses in a sense tagged corpus (i.e. SemCor), and WNf is calculated with the frequencies in WordNet. That represents bias of word senses in WordNet.

which is closely related to *crane*. Hence, the word *bird* can be a useful relative of the word *crane* though the word *bird* is ambiguous. But the ambiguous relative is not useful for other target words that are related to the least frequent senses of the relatives: that is, a relative *bird* is never helpful to disambiguate the senses of a word *birdie*, which is related to the least frequent sense of the relative *bird*.

We employ a weighting scheme for relatives in order to identify useful relatives for WSD. In terms of weights of relatives, our intent is to provide the useful relative with high weights, but the useless relatives with low weights. For instance, a relative *bird* of a word *crane* has a high weight whereas a relative *bird* of a word *birdie* get a low weight.

For the sake of the weights, we calculate similarities between a target word and its relatives and determine the weight of each relative based on the degree of the similarity. Among similarity measures between words, the total divergence to the mean (TDM) is adopted, which is known as one of the best similarity measures for word similarity [11].

Since TDM estimates a divergence between vectors, not between words, words have to be represented by vectors in order to calculate the similarity between the words based on the TDM. We define vector elements as words that occur more than 10 in a raw corpus, and build vectors of words by counting co-occurrence frequencies of the words and vector elements.

TDM does measure the divergence between words, and hence a reciprocal of the TDM measure is utilized as the similarity measure:

$$Sim(\vec{w_i}, \vec{w_j}) = \frac{1}{TDM(\vec{w_i}, \vec{w_j})}$$

where $Sim(\vec{w_i}, \vec{w_j})$ represents a similarity between two word vectors, $\vec{w_i}$ and $\vec{w_j}$.

A weight of a relative is determined by the similarity of a target word and its relative as follows:

$$W(r_{ij}, tw_i) = Sim(\vec{r_{ij}}, \vec{tw_i})$$

3.2   Co-occurrence Frequency Matrix

In order to select a relative for a target word in a given sentence, we must calculate probabilities of relatives given the sentence, as described in previous section. These probabilities as Eq. 5 and 6 can be estimated based on frequencies of relatives and co-occurrence frequencies between each relative and each word in the sentence.

In order to acquire the frequency information for calculating the probabilities, the previous relative based methods constructed a training data by collecting example sentences of relatives. However, to construct the training data requires a great amount of time and storage space. What is worse, it is an awful work to construct training datum of all ambiguous words, whose number is over than 20,000 in WordNet.

Instead, we build a co-occurrence frequency matrix (CFM) from a raw corpus that contains frequencies of words and word pairs. A value in the i-th row and

j-th column in the CFM represents the co-occurrence frequency of the i-th word and j-th word in a vocabulary, and a value in the i-th row and the i-th column represents the frequency of the i-th word.

The CFM is easily built by counting words and word pairs in a raw corpus. Furthermore, it is not necessary to make a CFM per each ambiguous word since a CFM contains frequencies of all words including relatives and word pairs. Therefore, our proposed method disambiguates senses of all ambiguous words efficiently by referring to only one CFM.

The frequencies in Eq. 5 and 6 can be obtained through a CFM as follows:

$$freq(w_i) = cfm(i, i) \tag{8}$$

$$freq(w_i, w_j) = cfm(i, j) \tag{9}$$

where w_i is a word, and $cfm(i, j)$ represents the value in the i-th row and j-th column of the CFM, in other word, the frequency that the i-th word and j-th word co-occur in a raw corpus.

4   Experiments

4.1   Experimental Environment

Experiments were carried out on several English sense tagged corpora: SemCor and corpora for both lexical sample task and all words task of both SENSEVAL-2 & -3.[5] SemCor [12][6] is a semantic concordance, where all content words (i.e. noun, verb, adjective, and adverb) are assigned to WordNet senses. SemCor consists of three parts: *brown1, brown2* and *brownv*. We used all of the three parts of the SemCor for evaluation.

In our method, raw corpora are utilized in order to build a CFM and to calculate similarities between words for the sake of the weights of relatives. We adopted Wall Street Journal corpus in Penn Treebank II [13] and LATIMES corpus in TREC as raw corpora, which contain about 37 million word occurrences.

Our CFM contains frequencies of content words and content word pairs. In order to identify the content words from the raw corpus, Tree-Tagger [14], which is a kind of automatic POS taggers, is employed.

WordNet provides various kinds of relationships between words or synsets. In our experiments, the relatives in Table 1 are utilized according to POSs of target words. In the table, *hyper3* means 1 to 3 hypernyms (i.e. parents, grandparents and great-grandparent) and *hypo3* is 1 to 3 hyponyms (i.e. children, grandchildren and great-grandchildren).

---

[5] We did not evaluate on verbs of lexical sample task of SENSEVAL-3 because the verbs are assigned to senses of WordSmyth, not WordNet.

[6] In this paper, SemCor 1.7.1 is adopted.

Table 1. Used Relative types

POS	relatives
noun	synonym, hyper3, hypo3, antonym, attribute, holonym, meronym, sibling
adjective	synonym, antonym, similar to, alsosee, attribute, particle, pertain
verb	synonym, hyper2, tropo2, alsosee, antonym, causal, entail, verbgroup
adverb	synonyms, antonyms, derived

4.2   Experimental Results

Comparison with Other Relative Based Methods. We tried to compare our proposed method with the previous relative based methods. However, both of [8] and [9] did not evaluate their methods on a publicly available data. We implemented their methods and compared our method with them on the same evaluation data.

When both of the methods are implemented, it is practically difficult to collect example sentences of all target words in the evaluation data. Instead, we implemented the previous methods to work with our CFM. WordNet was utilized as a lexical database to acquire relatives of target words and the sense disambiguation modules were implemented by using on Naïve Bayesian classifier, which [9] adopted though [8] utilized International Roget's Thesaurus and other classifier similar to decision lists. Also the bias of word senses, which is presented at WordNet, is reflected on the implementation in order to be in a same condition with our method. Hence, the reimplemented methods in this paper are not exactly same with the previous methods, but the main ideas of the methods are not corrupted. A correct sense of a target word tw_i in a sentence C is determined as follows:

$$
\begin{aligned}
&Sense(tw_i, C) \\
&\stackrel{\text{def}}{=} \arg\max_{s_{ij}} P(s_{ij}|C)P_{wn}(s_{ij})
\end{aligned}
\tag{10}
$$

where $Sense(tw_i, C)$ is a sense of tw_i in C, s_{ij} is the j-th sense of tw_i. $P_{wn}(s_{ij})$ is the WordNet probability of s_{ij}. The right hand side of Eq. 10 is calculated logarithmically under the assumption that words in C occur independently:

$$
\begin{aligned}
&\arg\max_{s_{ij}} P(s_{ij}|C)P_{wn}(s_{ij}) \\
&= \arg\max_{s_{ij}} \frac{P(C|s_{ij})P(s_{ij})}{P(C)} P_{wn}(s_{ij}) \\
&= \arg\max_{s_{ij}} P(C|s_{ij})P(s_{ij})P_{wn}(s_{ij}) \\
&= \arg\max_{s_{ij}} \{logP(C|s_{ij}) + logP(s_{ij})) \\
&+ logP_{wn}(s_{ij})\} \\
&\approx \arg\max_{s_{ij}} \{\sum_{k=1}^{n} logP(w_k|s_{ij}) + logP(s_{ij})) \\
&+ logP_{wn}(s_{ij})\}
\end{aligned}
\tag{11}
$$

where w_k is the k-th word in C and n is the number of words in C. In Eq. 11, we assume independence among the words in C.

Probabilities in Eq. 11 are calculated as follows:

$$P(w_k|s_{ij}) = \frac{P(s_{ij}, w_k)}{P(s_{ij})}$$
$$= \frac{freq(s_{ij}, w_k)}{freq(s_{ij})} \tag{12}$$

$$P(s_{ij}) = \frac{freq(s_{ij})}{CS} \tag{13}$$

$$P_{wn}(s_{ij}) = \frac{0.5 + WNf(s_{ij})}{n * 0.5 + WNf(tw_i)} \tag{14}$$

where $freq(s_{ij}, w_k)$ is the frequency that s_{ij} and w_k co-occur in a corpus, $freq(s_{ij})$ is the frequency of s_{ij} in a corpus, which is the sum of frequencies of all relatives related to s_{ij}. CS means corpus size, which is the sum of frequencies of all words in a corpus. $WNf(s_{ij})$ and $WNf(tw_i)$ are the frequencies of a s_{ij} and tw_i in WordNet, respectively, which represent bias of word senses. Eq. 14 is the same with Eq. 7 in Section 3.

Since the training data are built by collecting example sentences of relatives in the previous works, the frequencies in Eq. 12 and 13 are calculated with our matrix as follows:

$$freq(s_{ij}, w_k) = \sum_{r_l \ related \ to \ s_{ij}} freq(r_l, w_k)$$

$$freq(s_{ij}) = \sum_{r_l \ related \ to \ s_{ij}} freq(r_l)$$

where r_l is a relative related to the sense s_{ij}. $freq(r_l, w_k)$ and $freq(r_l)$ are the co-occurrence frequency between r_l and w_k and the frequency of r_l, respectively, and both frequencies can be obtained by looking up the matrix since the matrix contains the frequencies of words and word pairs.

The main difference between [8] and [9] is whether ambiguous relatives are utilized or not. Considering the difference, we implemented the method of [8] to include the ambiguous relatives into relatives, but the method of [9] to exclude the ambiguous relatives.

Table 2. Comparison results with previous relative-based methods

	S2 LS	S3 LS	S2 ALL	S3 ALL	SemCor
All Relatives	38.86%	42.98%	45.57%	51.20%	53.68%
Unambiguous Relatives	27.40%	24.47%	30.73%	33.61%	30.63%
our method	40.94%	45.12%	45.90%	51.35%	55.58%

Table 3. Comparison results with top 3 systems at SENSEVAL

	S2 LS	S2 ALL	S3 ALL
[15]	40.2%	56.9%	.
[16]	29.3%	45.1%	.
[5]	24.4%	32.8%	.
[17]	.	.	58.3%
[18]	.	.	54.8%
[19]	.	.	48.1%
Our method	40.94%	45.12%	51.35%

Table 2 shows the comparison results.[7] In the table, *All Relatives* and *Unambiguous Relatives* represent the results of the reimplemented methods of [8] and [9], respectively. It is observed in the table that our proposed method achieves better performance on all evaluation data than the previous methods though the improvement is not large. Hence, we may have an idea that our method handles relatives and in particular ambiguous relatives more effectively than [8] and [9].

Compared with [9], [8] obtains a better performance, and the difference between the performance of them are totally more than 15 % on all of the evaluation data. From the comparison results, it is desirable to utilize ambiguous relatives as well as unambiguous relatives.

[10] evaluated their method on nouns of lexical sample task of SENSEVAL-2. Their method achieved 49.8% recall. When evaluated on the same nouns of the lexical sample task, our proposed method achieved 47.26%, and the method of [8] 45.61%, and the method of [9] 38.03%. Compared with our implementations, [10] utilized a web as a raw corpus that is much larger than our raw corpus, and employed various kinds of features such as bigram, trigram, part-of-speeches, etc.[8] Therefore, it can be conjectured that a size of a raw corpus and features play an important role in the performance. We can observe that in our implementation of the method of [9], the data sparseness problem is very serious since unambiguous relatives are usually not frequent in the raw corpus. In the web, the problem seems to be alleviated. Further studies are required for the effects of various features.

---

[7] Evaluation measure is a *recall*, which is utilized for evaluating systems at SENSEVAL. In the table, S2 means SENSEVAL-2, LS means lexical sample task, and ALL represents all words task.

[8] [10] also utilized the bias information of word senses at WordNet.

Comparison with Systems Participated in SENSEVAL. We also compared our method with the top systems at SENSEVAL that did not use sense tagged corpora.[9] Table 3 shows the official results of the top 3 participating systems at SENSEVAL-2 & 3 and experimental performance of our method. In the table, it is observed that our method is ranked in top 3 systems.

5   Conclusions

We have proposed a simple and novel method that determines senses of all contents words in sentences by selecting a relative of the target words in WordNet. The relative is selected by using a co-occurrence frequency between the relative and the words surrounding the target word in a given sentence. The co-occurrence frequencies are obtained from a raw corpus, not from a sense tagged corpus that is often required by other approaches.

We tested the proposed method on SemCor data and SENSEVAL data, which are publicly available. The experimental results show that the proposed method effectively disambiguates many ambiguous words in SemCor and in test data for SENSEVAL all words task, as well as a small number of ambiguous words in test data for SENSEVAL lexical sample task. Also our method more correctly disambiguates senses than [8] and [9]. Furthermore, the proposed method achieved comparable performance with the top 3 ranked systems at SENSEVAL-2 & 3.

In consequence, our method has two advantages over the previous methods ([8] and [9]): our method 1) handles the ambiguous relatives and unambiguous relatives more effectively, and 2) utilizes only one co-occurrence matrix for disambiguating all contents words instead of collecting training data of the content words.

However, our method did not achieve good performances. One reason of the low performance is on the relatives irrelevant to the target words. That is, investigation of several instances which assign to incorrect senses shows that relatives irrelevant to the target words are often selected as the most probable relatives. Hence, we will try to devise a filtering method that filters out the useless relatives before the relative selection phase. Also we will plan to investigate a large number of tagged instances in order to find out why our method did not achieve much better performance than the previous works and to detect how our method selects the correct relatives more precisely. Finally, we will conduct experiments with various features such as bigrams, trigrams, POSs, etc, which [10] considered and examine a relationship of a size of a raw corpus and a system performance.

---

[9] At SENSEVAL, unsupervised systems include the weakly supervised systems though there are some debates. In this paper, our methods are compared with the systems that are classified into the unsupervised approach at SENSEVAL.

References

1. Fellbaum, C.: An WordNet Electronic Lexical Database. The MIT Press (1998)
2. Schütze, H.: Automatic word sense discrimination. Computational Linguistics **24** (1998) 97–123
3. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the 5th annual international conference on Systems documentation, Toronto, Ontario, Canada (1986) 24–26
4. Karov, Y., Edelman, S.: Similarity-based word sense disambiguation. Computational Linguistics **24** (1998) 41–59
5. Haynes, S.: Semantic tagging using WordNet examples. In: Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, France (2001) 79–82
6. Agirre, E., Rigau, G.: Word sense disambiguation using conceptual density. In: Proceedings of COLING'96, Copenhagen Denmark (1996) 16–22
7. Fernandez-Amoros, D., Gonzalo, J., Verdejo, F.: The role of conceptual relations in word sense disambiguation. In: Proceedings of the 6th International Workshop on Applications of Natural Language for Information Systems, Madrid, Spain (2001) 87–98
8. Yarowsky, D.: Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In: Proceedings of COLING-92, Nantes, France (1992) 454–460
9. Leacock, C., Chodorow, M., Miller, G.A.: Using corpus statistics and WordNet relations for sense identification. Computational Linguistics **24** (1998) 147–165
10. Agirre, E., Martinez, D.: Unsupervised wsd based on automatically retrieved examples: The importance of bias. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Barcelona, Spain (2004)
11. Lee, L.: Similarity-Based Approaches to Natural Language Processing. PhD thesis, Harvard University, Cambridge, MA (1997)
12. Miller, G.A., Leacock, C., Tengi, R., Bunker, R.: A semantic concordance. In: Proceedings of the 3 DARPA Workshop on Human Language Technology. (1993) 303–308
13. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: The penn treebank. Computational Linguistics **19** (1994) 313–330
14. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the Conference on New Methods in Language Processing, Manchester, UK (1994)
15. Fernandez-Amoros, D., Gonzalo, J., Verdejo, F.: The UNED systems at SENSEVAL-2. In: Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, France (2001) 75–78
16. Litkowski, K.: SENSEVAL-2:overview. In: Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, France (2001) 107–110

17. Strapparava, C., Gliozzo, A., Giuliano, C.: Pattern abstraction and term similarity for word sense disambiguation: Irst at senseval-3. In: Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain (2004) 229–234
18. Fernandez-Amoros, D.: Wsd based on mutual information and syntactic patterns. In: Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain (2004) 117–120
19. Buscaldi, D., Rosso, P., Masulli, F.:   The upv-unige-ciaosenso wsd system. In: Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain (2004) 77–82