Word Sense Disambiguation Using Neural Networks with Concept Co-occurrence Information

You-Jin Chung, Sin-Jae Kang, Kyong-Hi Moon, and Jong-Hyeok Lee
Div. of Electrical and Computer Engineering
Pohang University of Science and Technology (POSTECH)
San 31, Hyoja-dong, Nam-gu, Pohang, 790-784, R. of KOREA
{prizer,sjkang,khmoon,jhlee}@postech.ac.kr

Abstract

Most previous word sense disambiguation approaches based on neural networks were impractical due to their huge feature set size. We propose a method for resolving word sense ambiguity using neural networks with refined concept co-occurrence information (CCI) as features. Using CCI refinement processing, we reduce the number of features of the network to a practical size. We also show that word sense disambiguation can be improved by combining several clues rather than using them independently. Our method is fully automated and does not require any hand coding of large-scale resources.

1 Introduction

In Korean-to-Japanese machine translation (MT), employing a direct MT strategy, a Korean homograph may be translated into a different Japanese equivalent depending on which sense is used in a given context. Thus, word sense disambiguation (WSD) is essential to the selection of an appropriate Japanese target word and has been a major interest and concern in MT.

Much research has been done on word sense disambiguation. Researchers have found that several different kinds of information can contribute to the resolution of lexical ambiguity. These include surrounding words (unordered set of words surrounding a target word), local collocations (short sequence of words near a target word, taking word order into account), syntactic relations (selectional restrictions), parts of speech, morphological forms, etc (McRoy, 1992, Ng and Zelle, 1997).

Li *et al.* (2000) suggested a corpus-based method that uses concept co-occurrence information (CCI), such as local syntactic patterns and unordered co-occurrence words. He automatically extracted CCI from a sense-tagged corpus, and constructed a four-step algorithm. Their method considers only one clue at each decision step rather than several clues together. It happens that the method fails to disambiguate by the first clue, although the second clue can make the right decision. Thus, if the combined clues are used, the method achieves a better performance.

Some approaches to word sense disambiguation use neural networks. Waltz *et al.* (1985) and Gallant (1991) proposed a neural network classifier using semantic microfeatures. Since their method requires large amounts of hand-written data, it is not clear that the same neural net models will scale up for realistic application.

Leacock *et al.* (1993) and Mooney (1996) used a few thousand words as a feature set of neural networks. The input patterns are composed of a few thousand binary features, each representing the presence or absence of a particular word stem in the context of an input sentence. Due to their huge feature set size, however, it is impractical to apply their models to real world applications.

We propose a method for word sense disambiguation that combines both the neural net-based approach and the work of Li *et al.* We focus especially on the construction of the refined feature set of a practical size. This is achieved by a CCI refinement processing, such as concept discrimination and concept code generalization. Unlike previous neural network approaches, our method is fully automated and does not require any hand coding of large-scale resources. To improve the applicability of the method, we adopt a concept similarity

calculation scheme rather than an exact matching scheme.

This paper is organized as follows. Section 2 describes the automatic construction of a sense-tagged Korean corpus and the extraction of refined features from them. Section 3 explains the network architecture and our sense disambiguation method. In Section 4, the experimental results are given, showing that the proposed method may be useful for WSD in a real text. The concluding remarks are given in Section 5. In this paper, Yale Romanization is used to represent Korean expressions.

2 Construction of the Refined Feature Set

For practical reasons, a resonably small number of features is very important in the design of neural network. To make a feature set of reasonable size, we adopt Li's method (2000), based on concept co-occurrence information.

The architecture of the feature extraction system is shown in Figure 1. A feature set for the neural network is composed of two kinds of concept co-occurrence information (CCI) : local syntactic patterns (LSPs) and unordered co-occurrence words (UCWs). CCI are concept codes in a thesaurus.

To extract LSPs and UCWs from a corpus

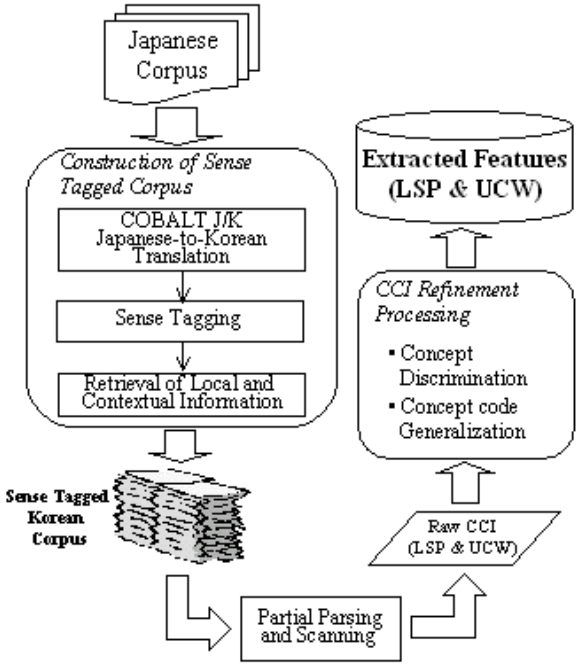

Figure 1. Extraction of the refined features

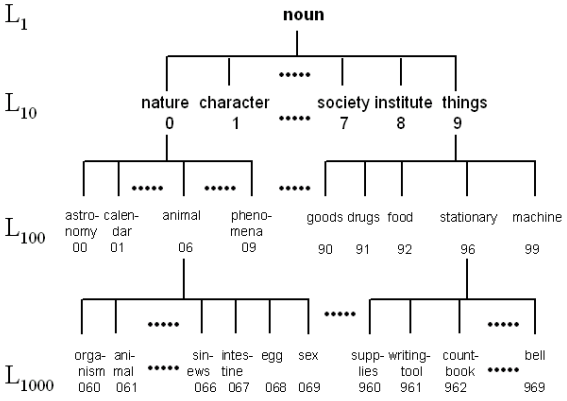

Figure 2. Concept hierarchy of the Kadokawa thesaurus

automatically, following the Li's method, we constructed a sense-tagged corpus using a Japanese-to-Korean MT system at first, and then extracted LSPs and UCWs from them through partial parsing and scanning. To reduce the number of concept codes in LSPs and UCWs, they are then filtered and later code generalized through statistical processing. After the refinement processing is completed, the remained LSPs and UCWs are used as features for the neural network in Section 3.

2.1 Automatic Construction of the Sense-tagged Corpus

For automatic construction of the sense-tagged corpus, we used a Japanese-to-Korean MT system called COBALT-J/K[1] (Park *et al.*, 1997). In the transfer dictionary of COBALT-J/K, nominal and verbal words are annotated with concept codes of the Kadokawa thesaurus (Ohno and Hamanishi, 1981), which has a 4-level hierarchy of about 1,100 semantic classes, as shown in Figure 2. Concept nodes in level L_1, L_{10} and L_{100} are further divided into 10 subclasses.

We made slight modification of COBALT-J/K so that it can produce Korean translations from a Japanese text with all nominal words tagged with specific concept codes at level L_{1000} of the Kadokawa thesaurus. As a result, a Korean sense-tagged corpus can be obtained from Japanese texts.

[1] COBALT-J/K is a high-quality practical MT system developed by POSTECH (Pohang University of Science and Technology) in 1996.

Table 1. Local collocation patterns (LSPs)

LSP type	Structure of collocation
$type_1$	**noun** + noun
$type_2$	noun + *uy* + **noun**
$type_3$	**noun** + *other particles* + noun
$type_4$	**noun** + *lo/ulo* + verb
$type_5$	**noun** + *ey* + verb
$type_6$	**noun** + *eygey* + noun
$type_7$	**noun** + *eyse* + verb
$type_8$	**noun** + *ul/lul* + verb
$type_9$	**noun** + *i/ka* + verb
$type_{10}$	verb + *relativizer* + **noun**

In the automatic construction of a sense-tagged corpus, quality of the corpus is a critical issue. To examine the quality of the sense-tagged corpus, we collected 878 sample sentences (16,527 eojeols[2]) from the corpus and then checked their accuracy. The total number of errors was 382, including morphological analysis errors, sense ambiguity resolution errors and unknown words errors. It corresponds to the accuracy of 97.7% (16145 / 16527 eojeols).

Because almost all Japanese common nouns represented by Chinese characters have only one sense, there is little ambiguity in Japanese-to-Korean translation. In our test, the number of ambiguity resolution errors was 81 and it took only 0.49% of the overall corpus (81 / 16527 eojeols). Considering the fact that the overall accuracy of the constructed corpus is over 97% and only a few sense ambiguity resolution errors were found in the Japanese-to-Korean translation of nouns, we regard the generated sense-tagged corpus as highly reliable.

2.2 Extraction of LSPs and UCWs

Unlike English, Korean has almost no syntactic constraints on word order as long as the verb appears in the final position. The variable word order often results in discontinuous constituents. Instead of using local collocations by word order, Li *et al.* (2000) defined 12 local syntactic patterns (LSPs) for homographs using syntactically related words in a sentence. We adopt only 10 patterns among them, as shown in Table 1.

For a homograph W, concept frequency patterns (CFPs), i.e., ($\{<C_1,f_1>,<C_2,f_2>, \ldots, <C_k,f_k>\}$, $type_i$, $W(S_i)$), are extracted for each type i of LSP by partial parsing and scanning, where k is the number of concept codes in $type_i$, f_i is the frequency of concept C_i appearing in the corpus, $type_i$ is an LSP type i, and $W(S_i)$ is a homograph W with a sense S_i.

Frequently co-occurring words in a sentence are retrieved as unordered co-occurring words (UCWs). They are not included in LSPs for the homographs. CFPs extracted for UCWs referring to co-occurred frequency are ($\{<C_1,f_1>, <C_2,f_2>, \ldots, <C_k,f_k>\}$, UCW, $W(S_i)$).

2.3 Concept Discrimination of CCI

In the extracted LSPs and UCWs, however, the same concept may appear for determining the different meanings of a homograph. To select the most probable concept codes, which frequently co-occur with the target sense of a homograph, from the extracted CCI, Li used Shannon's entropy (Shannon, 1951) to define the noise of a certain concept codes for discrimination of ambiguous word senses.

Let S_i represent the i^{th} word sense of homograph W, C_k, the concept code of the co-occurring word, $p(C_k|S_i)$, the probability that represents the possibility that the concept C_k will co-occur with the word sense S_i in a sentence, and n be the number of word senses of W. The noise generated by concept code C_k is defined as Equation 1 and the discrimination value DS_k of concept code C_k for W is defined as Equation 2.

$$noise_k = -\sum_{i=1}^{n} \frac{p(C_k \mid S_i)}{\sum_{j=1}^{n} p(C_k \mid S_j)} \log_2 \frac{p(C_k \mid S_i)}{\sum_{j=1}^{n} p(C_k \mid S_j)} \quad (1)$$

$$DS_k = \frac{\log_2 n - noise_k}{\log_2 n} \quad (2)$$

If the discrimination value DS_k of the concept code C_k is larger than the threshold DS_{th}, the concept code is selected as useful information for deciding word sense S_i. Otherwise, the concept code is discarded.

2.4 Concept Code Generalization

After the concept discrimination, co-occurring concept codes in each LSP, as well as UCWs, are needed to be further selected and the code generalized. For the purpose of code selection and generalization of concepts in LSPs and

[2] Eojeol is a spacing unit of Korean.

Table 2. Concept codes and frequencies in CFP ({<C_i,f_i>}, $type_2$, *nwun*(eye))

Code	Freq.	Code	Freq.	Code	Freq.	Code	Freq.
103	4	107	8	121	7	126	4
143	8	160	5	179	7	277	4
320	8	331	6	416	7	419	12
433	4	501	13	503	10	504	11
505	6	507	12	508	27	513	5
530	6	538	16	552	4	557	7
573	5	709	5	718	5	719	4
733	5	819	4	834	4	966	4
987	9	other*	210				

※ 'other' in the table means the set of concept codes with the frequencies less than 4.

UCWs, the Kadokawa thesaurus is used. All concepts in LSPs and UCWs are three-digit concept codes at level L_{1000} in the Kadokawa thesaurus. Table 2 shows the concept codes that can co-occur with the homograph '*nwun*(eye)' in the form of LSP type$_2$ and their frequencies.

To perform code generalization, Li *et al.* refered to Smadja's work (Smadja, 1990, 1993), and defined the standard deviation σ_ℓ of the code frequency at the thesaurus level ℓ (denoted as L_ℓ) and $k_{f,\ell}$ as in Equation 3 and 4. $k_{f,\ell}$ is the strength of code frequency f at L_ℓ, which represents the amount of standard deviation above the average frequency $f_{ave,\ell}$. In the equations, $f_{k,\ell}$ denotes the frequency of concept code C_k of the Kadokawa thesaurus at L_ℓ, and n_ℓ the number of concept codes at L_ℓ.

$$\sigma_\ell = \sqrt{\frac{\sum_{k=1}^{n_\ell} (f_{k,\ell} - f_{ave,\ell})^2}{n_\ell - 1}} \quad (3)$$

$$k_{f_{k,\ell},\ell} = \frac{f_{k,\ell} - f_{ave,\ell}}{\sigma_\ell} \quad (4)$$

The generalization filter of the system selects the concept codes with a variation σ_ℓ larger than threshold $\sigma_{th,\ell}$, and pulls out the concept codes with a strength of frequency $k_{f,\ell}$ larger than threshold $k_{th,\ell}$. If the value of σ_ℓ is small, then it can be assumed that no peak frequency of the code for the pattern exists.

After generalization at L_{1000}, the system performs the same work at L_{100}. It assigns a zero value to the frequencies of codes that are selected at L_{1000}, and sums up the frequencies of the remaining concept codes to form CFPs with codes of a higher level of the concept hierarchy. After processing, the system selects the most promising codes and stores the conceptual patterns ({C_1, C_2, C_3, ...}, $type_i$, $W(S_i)$) as a knowledge source for WSD of real texts. The code generalized LSP for type$_2$ of noun *nwun*(eye) in Table 2 is ({028, 419, 501, 504, 507, 508, 538, 50}, $type_2$, *nwun*(eye)). For UCWs, the same processing is applied. The generalized LSPs and UCWs are used as features for the neural network. The more specific description of the extraction of LSPs and UCWs are explained in Li (2000).

3 Network Architecture and Sense Disambiguation

3.1 Neural Network Architecture

Due to its strong capability for classification, the multilayer feedforward neural network with a backpropagation learning algorithm is used in our sense classification system. As shown in Figure 3, each node in the input layer represents a concept code in CCI (LSPs & UCWs) of a target word and each node in the output layer represents a sense of a target word. The number of hidden layers and the number of nodes in a hidden layer are a critical issue. To determine a good topology for the network, we implemented 2 layered (no hidden layer) and 3 layered (with a single hidden layer of 5 nodes) network. Each layer in the network is fully connected to the next layer.

Each homograph has a network of its own.

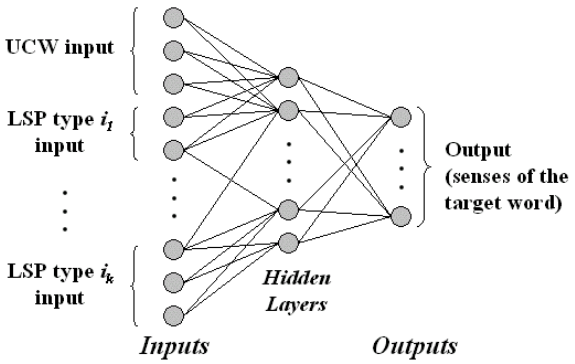

Figure 3. Topology of the network

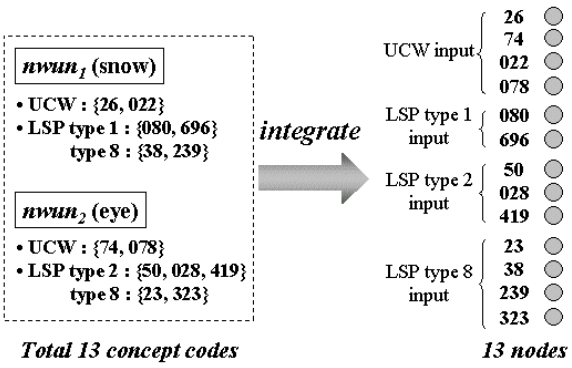

Figure 4. Construction of the input layer for '*nwun*'

Figure 4[3] shows a construction example of the input layer for the homograph '*nwun*' with the sense 'snow' and 'eye'. The sense 'snow' has concept code 26, 022, 080, 696, 38 and 239 as its CCI and the sense 'eye' has 74, 078, 50, 028, 419, 23 and 323. We make the input layer for '*nwun*' by integrating the concept codes in both senses and the resulted input layer is partitioned into several subgroups depending on their CCI types, i.e., UCW, LSP type 1, LSP type 2 and LSP type 8.

3.2 Network Training

To generate a sense tagged corpus and to extract LSPs and UCWs for each homograph, we used the COBALT-J/K system described in Section 2.1. and a corpus which composed of 240,000 sentences from corpus of EDR electronic dictionary, Asahi Newspaper, and Japanese Newspaper of Economics. Using the extracted LSPs and UCWs, we constructed neural networks and trained network parameters for each homograph. The average number of input nodes was about 35. If we assume that the average number of senses of homographs is 2, the total number of network parameters (synaptic weights) for each homograph is 70 (35×2) in the case of a 2-layered network, which is a very reasonable size to be used for real world applications. In the case of a 3-layered network with 5 hidden nodes, the total number of parameters is 185 (35×5 + 5×2).

3.3 Sense Disambiguation

[3] The concept codes in Figure 4 are simplified ones for the ease of illustration. In reality there are 87 concept codes for '*nwun*'.

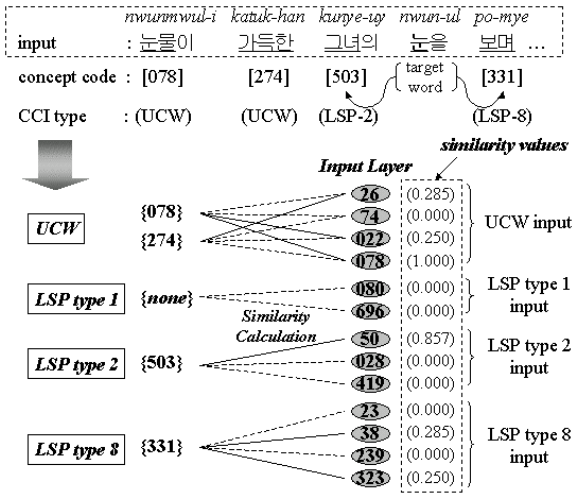

Figure 5. Construction of the input pattern by using a concept similarity calculation

For a given homograph *W*, the sense disambiguation is performed by the following three steps.

Step 1. Extract LSPs and UCWs from the context of *W*. The window size of the UCW extraction is a single sentence. Consider, for example, the sentence in Figure 5 which has the meaning of "Seeing her eyes filled with tears, …". The target word is the homograph '*nwun*'. We extract its LSPs and UCWs from the sentence by partial parsing and scanning. In Figure 5, for example, the words '*nwun*' and '*kunye*' with the concept code 503 have the relation of <noun + *uy* + noun> and it corresponds to LSP type 2. There is no syntactic relation between the words *'nwun'* and '*nwunmul*' with the concept code 078, so we assign '*UCW'* as the CCI-type of the concept code 078.

In a similar manner, we can obtain all pairs of the CCI type and their concept codes appearing in the context. All the extracted <*CCI-type*: *concept codes*> pairs are as follows: {<*UCW*: *078,274*>, <*LSP type 2*: *503*>, <*LSP type 8*: *331*>}.

Step 2. Obtain the input pattern by calculating concept similarities between the features of the input nodes and the concept code in the extracted <*CCI-type: concept codes*>. Similarity calculation is performed only between the concept codes with the same CCI-type. The calculated concept similarity is assigned to each input node as the activation value to the network.

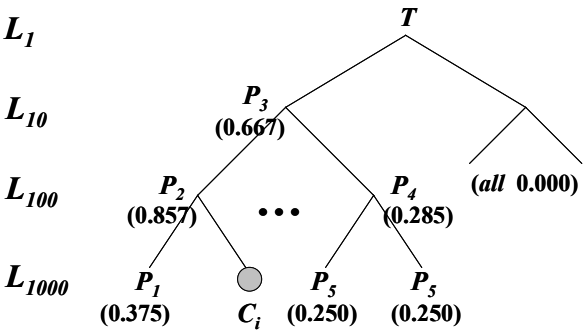

Figure 6. Concept similarity on the Kadokawa thesaurus hierarchy

$Csim(C_i, P_j)$ in Equation 5 is used to calculate the concept similarity between C_i and P_j, where $MSCA(C_i, P_j)$ is the most specific common ancestor of concept codes C_i and P_j, and *weight* is a weighting factor reflecting that C_i as a descendant of P_j is preferable to other cases. The similarity values between C_i and each P_j on the Kadokawa thesaurus hierarchy are shown in Figure 6.

$$Csim(C_i, P_j) = \frac{2 \times level(MSCA(C_i, P_j))}{level(C_i) + level(P_j)} \times weight \quad (5)$$

For example, in UCW part calculation, the relation between the concept codes 274 and 26 corresponds to the relation between C_i and P_4 in Figure 6. So we assign the similarity 0.285 to the input node labeled by 26. If there more than two concept codes exist in one CCI-type such as <*UCW*: *078, 274*>, the maximum similarity value among them is assigned to the input node as Equation 6.

$$FeatureVal(C_i) = \max_{P_j}(Csim(C_i, P_j)) \quad (6)$$

In Equation 6, C_i is the concept code of the input node, and P_j is the concept codes in the <*CCI-type*: *concept codes*> pair which has the same CCI-type with C_i.

By adopting this concept similarity calculation, we can achieve a broad applicability of the method. If we use the exact matching scheme instead of the concept similarity scheme, there would be only a few concept codes matched with the features. Consequently, sense disambiguation would fail because of the absence of clues.

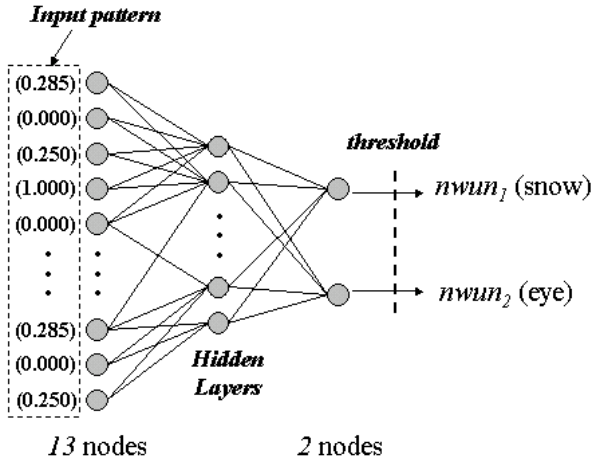

Figure 7. Sense disambiguation of '*nwun*'

Step 3. Feed the obtained input pattern to the network and compute activation strengths for each output sense node (Figure 7). Next, select the sense of the node that has a larger activation value than any other output node. If the output value is lower than the threshold, it will be discarded and the network will not make any decisions.

4 Experimental Evaluation

For an experimental evaluation, eight ambiguous Korean nouns were selected, along with a total of 3514 test sentences in which one of the homographs appears. The test sentences were randomly selected from the Korean sense-tagged corpus generated by the method in Section 2.1, which were not used for the training of networks. Out of several senses for each homograph, we considered only two senses that are most frequently used in the corpus. For each sense of a homograph, the number of its appearances in the test sentences is shown in Table 3.

We performed three experiments. The first experiment, LSP_UCW, is the case where LSPs and UCWs are used independently, following Li's method. In this method, LSPs are used initially as clues, and if the maximum score of LSPs is smaller than a threshold, UCWs are used next. We set the threshold at 0.3.

Acc and *App* in the table indicate accuracy and applicability respectively and they are defined as in Equation 7 and 8, where $N_{correct}$ is the number of correctly disambiguated instances, $N_{applied}$ is the number of instances which the method

Table 3. Experimental results of word sense disambiguation (%)

Homograph	Sense	No	LSP_UCW		2-layered NN		3-layered NN	
			Acc	*App*	*Acc*	*App*	*Acc*	*App*
Pwuca	father & child	330	66.5	69.7	78.8	80.8	75.9	93.1
	rich man	165						
Kancang	liver	142	72.1	98.6	70.0	96.2	71.6	96.7
	soy sauce	69						
Kasa	housework	456	65.1	81.2	85.2	90.7	83.9	93.6
	words of song	103						
Kwutwu	shoes	651	76.4	81.0	97.2	99.4	97.3	99.0
	word of mouth	19						
Nwun	eye	364	64.8	58.5	86.9	97.8	85.6	96.9
	snow	87						
Yongki	courage	95	68.5	94.3	85.9	92.4	80.1	95.6
	container	467						
Uysa	intention	184	86.5	88.9	91.2	85.4	90.4	88.0
	doctor	267						
Cikwu	the earth	285	79.4	81.8	84.7	83.8	85.0	87.0
	district	160						
Average			72.4	81.8	85.0	90.8	83.7	93.7

applied to, and N_{total} is the total number of instances.

$$Accuracy = N_{correct} / N_{applied} \quad (7)$$

$$Applicability = N_{applied} / N_{total} \quad (8)$$

The second and the third experiment in Table 3 show the results of our WSD methods using 2-layered (with no hidden layer) and 3-layered neural networks (with 5 hidden nodes) independently. The threshold for the output activation function was fixed at 0.6 in both models.

The 2-layered and 3-layered neural network models have achieved a 12.6% and 11.3% improvement in accuracy and a 9.0% and 11.9% improvement in applicability over the LSP_UCW method, respectively. This result shows that word sense disambiguation can be improved by combining several clues together rather than using them independently.

Comparing the 2-layered and the 3-layered neural networks, similar performance was obtained. However, the 2-layered model has less parameters to adjust, 70 to 185, so that it is more efficient to generalize for limited training corpora.

5 Conclusion

To resolve sense ambiguities in Korean-to-Japanese MT, this paper has proposed a word sense disambiguation method using neural networks with refined concept co-occurrence information (CCI) as features. CCI consists of two types of information : local syntactic patterns and unordered co-occurrence words. Using CCI refinement processing, we can reduce the number of features of the network to a practical size. Unlike most previous approaches based on neural networks, our method is fully automated and requires only a small number of features. In an experimental evaluation, the proposed WSD model using 2-layered network achieved the average accuracy of 85.0% with an improvement over Li's method by 12.6%. Because the test corpus is completely irrelevant to the training corpus, the model may be effective for noun-sense disambiguation.

We plan further research to improve accuracy and applicability, and to expand our method for verb homograph disambiguation.

Acknowledgements

The authors would like to thank the Ministry of Education of Korea for its financial support toward the Electrical and Computer Engineering

Division at POSTECH through its BK21 program.

References

Gallant, Stephen I. 1991. A Practical Approach for Representing Context and for Performing Word Sense Disambiguation Using Neural Networks, Neural Computation, 3(3): 293-309

Leacock, Claudia, Geoffrey Towell and Ellen Voorhees. 1993. *Corpus-based Statistical Sense Resolution,* In Proceedings of the ARPA Human Language Technology Workshop, San Francisco, Morgan Kaufman: pp.260-265

Li, Hui-Feng, Namwon Heo, Kyounghi Moon, Jong-Hyeok Lee and Geunbae Lee. 2000. *Lexical Transfer Ambiguity Resolution Using Automatically-Extracted Concept Co-occurrence Information,* International Journal of Computer Processing of Oriental Languages, 13(1): 53-68

McRoy, Susan W. 1992. Using Multiple Knowledge Sources for Word Sense Discrimination, Computational Linguistics, 18(1): 1-30

Mooney, Raymond J. 1996. *Comparative Experiments on Disambiguating Word Senses: An Illustration of the Role of Bias in Machine Learning,* In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA: pp.82-91

Ng, Hwee Tou and John Zelle. 1997. Corpus-Based Approaches to Semantic Interpretation in Natural Language Processing, AI Magazine, 18(4): 45-64

Ohno, Susumu and Masando Hamanishi. 1981. *New Synonym Dictionary*, Kadokawa Shoten, Tokyo (in Japanese)

Park, Chul-Jae, Jong-Hyeok Lee, Geunbae Lee and K. Kakechi. 1997. *Collocation-Based Transfer Method in Japanese-Korean Machine Translation*, Transaction of Information Processing Society of Japan, 38(4): 707-718 (in Japanese)

Smadja, Frank. 1990. *Automatically Extracting and Representing Collocations for Language Generation*, in Proceedings of 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh, Pennsylvania: pp.252-259

Smadja, Frank. 1993. *Retrieving Collocations from Text: Xtract*, Computational Linguistics, 19(1): 143-177

Waltz, David L. and Jordan Pollack. 1985. Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation, Cognitive Science, 9: 51-74