Word Sense Disambiguation in a Korean-to-Japanese MT System Using Neural Networks

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)
and Advanced Information Technology Research Center(AlTrc)
San 31, Hyoja-dong, Nam-gu, Pohang, R. of KOREA, 790-784
{prizer,sjkang,khmoon,jhlee}@postech.ac.kr

Abstract

This paper presents a method to resolve word sense ambiguity in a Korean-to-Japanese machine translation system using neural networks. The execution of our neural network model is based on the concept codes of a thesaurus. Most previous word sense disambiguation approaches based on neural networks have limitations due to their huge feature set size. By contrast, we reduce the number of features of the network to a practical size by using concept codes as features rather than the lexical words themselves.

Introduction

Korean-to-Japanese machine translation (MT) employs a direct MT strategy, where 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.

Much research on word sense disambiguation has revealed that several different types of information can contribute to the resolution of lexical ambiguity. These include surrounding words (an unordered set of words surrounding a target word), local collocations (a 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).

Some researchers use neural networks in their word sense disambiguation systems Because of its strong capability in classification (Waltz *et al.,* 1985, Gallant, 1991, Leacock *et al.*, 1993, and Mooney, 1996). Since, however, most such methods require a few thousands of features or large amounts of hand-written data for training, it is not clear that the same neural network models will be applicable to real world applications.

We propose a word sense disambiguation method that combines both the neural net-based approach and the work of Li *et al* (2000), especially focusing on the practicality of the method for application to real world MT systems. To reduce the number of input features of neural networks to a practical size, we use concept codes of a thesaurus as features.

In this paper, Yale Romanization is used to represent Korean expressions.

1 System Architecture

Our neural network method consists of two phases. The first phase is the construction of the feature set for the neural network; the second phase is the construction and training of the neural network. (see Figure 1.)

For practical reasons, a reasonably small number of features is essential to the design of a neural network. To construct a feature set of a reasonable size, we adopt Li's method (2000), based on concept co-occurrence information (CCI). CCI are concept codes of words which co-occur with the target word for a specific syntactic relation.

In accordance with Li's method, we automatically extract CCI from a corpus by constructing a Korean sense-tagged corpus. To accomplish this, we apply a Japanese-to-Korean MT system. Next, we extract CCI from the constructed corpus through partial parsing and scanning. To eliminate noise and to reduce the number of CCI, refinement proceesing is applied

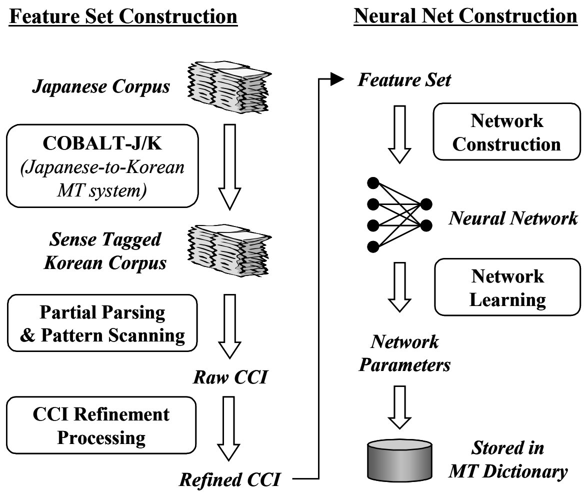

Figure 1. System Architecture

to the extracted raw CCI. After completing refinement processing, we use the remaining CCI as features for the neural network. The trained network parameters are stored in a Korean-to-Japanese MT dictionary for WSD in translation.

2 Construction of Refined Feature Set

2.1 Automatic Construction of Sense-tagged Corpus

For automatic construction of the sense-tagged corpus, we used a Japanese-to-Korean MT system called COBALT-J/K[1]. 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_2 and L_3 are further divided into 10 subclasses.

We made a slight modification of COBALT-J/K to enable it to produce Korean translations from a Japanese text, with all nominal words tagged with specific concept codes at level L_4 of the Kadokawa thesaurus. As a result, a Korean sense-tagged corpus of 1,060,000 sentences can be obtained from the Japanese corpus (*Asahi Shinbun*, Japanese Newspaper of Economics, etc.).

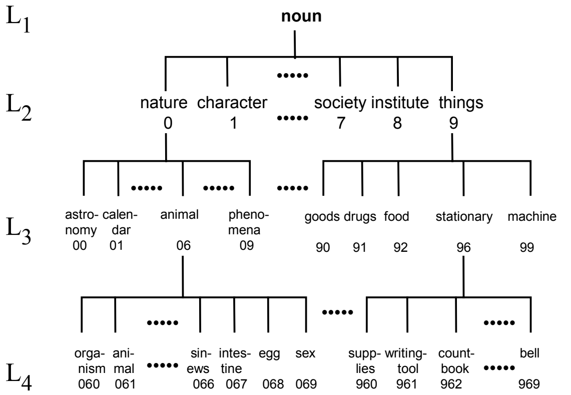

Figure 2. Concept hierarchy of the Kadokawa thesaurus

The quality of the constructed sense-tagged corpus is a critical issue. To evaluate the quality, we collected 1,658 sample sentences (29,420 eojeols[2]) from the corpus and checked their precision. The total number of errors was 789, and included such errors as morphological analysis, sense ambiguity resolution and unknown words. It corresponds to the accuracy of 97.3% (28,631 / 29,420 eojeols).

Because almost all Japanese common nouns represented by Chinese characters are monosemous little transfer ambiguity is exhibited in Japanese-to-Korean translation. In our test, the number of ambiguity resolution errors was 202 and it took only 0.69% of the overall corpus (202 / 29,420 eojeols). Considering the fact that the overall accuracy of the constructed corpus exceeds 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 Raw CCI

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 13 patterns of CCI for homographs using syntactically related words in a sentence. Because we are concerned only with

[1] COBALT-J/K (Collocation-Based Language Translator from Japanese to Korean) is a high-quality practical MT system developed by POSTECH.

[2] An Eojeol is a Korean syntactic unit consisting of a content word and one or more function words.

Table 1. Structure of CCI Patterns

CCI type	Structure of pattern
$type_0$	unordered co-occurrence words
$type_1$	**noun** + noun or 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* + verb
$type_7$	**noun** + *eyse* + verb
$type_8$	**noun** + *ul/lul* + verb
$type_9$	**noun** + *i/ka* + verb
$type_{10}$	verb + relativizer + **noun**

noun homographs, we adopt 11 patterns from them excluding verb patterns, as shown in Table 1. The words in bold indicate the target homograph and the words in italic indicate Korean particles.

For a homograph *W*, concept frequency patterns (CFPs), i.e., ($\{<C_1,f_1>,<C_2,f_2>, ... , <C_kf_k>\}$, $type_i$, $W(S_i)$), are extracted from the sense-tagged training corpus for each CCI type *i* by partial parsing and pattern scanning, where *k* is the number of concept codes in $type_i$, f_i is the frequency of concept code C_i appearing in the corpus, $type_i$ is an CCI type *i*, and $W(S_i)$ is a homograph *W* with a sense S_i. All concepts in CFPs are three-digit concept codes at level L_4 in the Kadokawa thesaurus. Table 2 demonstrates an example of CFP that can co-occur with the homograph '*nwun*(eye)' in the form of the CCI $type_2$ and their frequencies.

2.3 CCI Refinement Processing

The extracted CCI are too numerous and too noisy to be used in a practical system, and must to be further selected. To eliminate noise and to reduce the number of CCI to a practical size, we apply the refinement processing to the extracted CCI. CCI refinement processing is composed of 2 processes: concept code discrimination and concept code generalization.

2.3.1 Concept Code Discrimination

In the extracted CCI, the same concept code 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, Li defined the discrimination value of a concept code using Shannon's entropy (Shannon, 1951). A concept code with a small entropy has a large discrimination value. If the discrimination value of the concept code is larger than a threshold, the concept code is selected as useful information for deciding the word sense. Otherwise, the concept code is discarded.

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.

2.3.2 Concept Code Generalization

After concept discrimination, co-occurring concept codes in each CCI type must be further selected and the code generalized. To perform code generalization, Li adopted to Smadja's work (Smadja, 1993) and defined the code strength using a code frequency and a standard deviation in each level of the concept hierarchy. The generalization filter selects the concept codes with a strength larger than a threshold. We perform this generalizaion processing on the Kadokawa thesaurus level L_4 and L_3.

After processing, the system stores the refined conceptual patterns ($\{C_1, C_2, C_3, ...\}$, $type_i$, $W(S_i)$) as a knowledge source for WSD of real texts. These refined CCI are used as input features for the neural network. The more specific description of the CCI extraction is explained in Li (2000).

3 Construction of Neural Network

3.1 Neural Network Architecture

Because of its strong capability for classification, the multilayer feedforward neural network is used in our sense classification system. As shown in Figure 3, each node in the input layer represents a concept code in CCI of a target

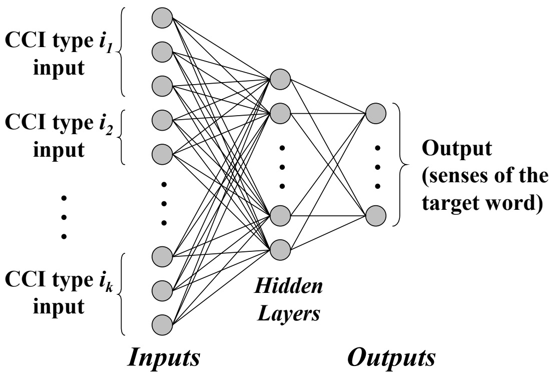

Figure 3. Topology of Neural Network

word and each node in the output layer represents the sense of a target word. The number of hidden layers and the number of nodes in a hidden layer are another crucial issue. To determine a good topology for the network, we implemented a 2-layer (no hidden layer) and a 3-layer (with a single hidden layer of 5 nodes) network and compared their performance. The comparison result is given in Section 5.

Each homograph has a network of its own. Figure 4[3] demonstrates a construction example of the input layer for the homograph '*nwun*' with the sense 'snow' and 'eye'. The left side is the extracted CCI for each sense after refinement processing. We construct the input layer for '*nwun*' by merely integrating the concept codes in both senses. The resulting input layer is partitioned into several subgroups depending on their CCI types, i.e., type 0, type 1, type 2 and type 8. Figure 5 shows the overall network architecture for '*nwun*'.

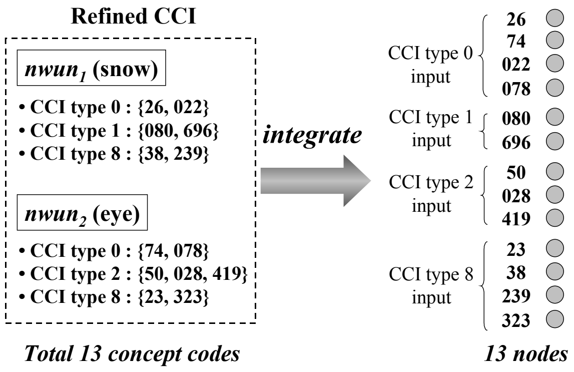

Figure 4. Construction of Input layer for '*nwun*'

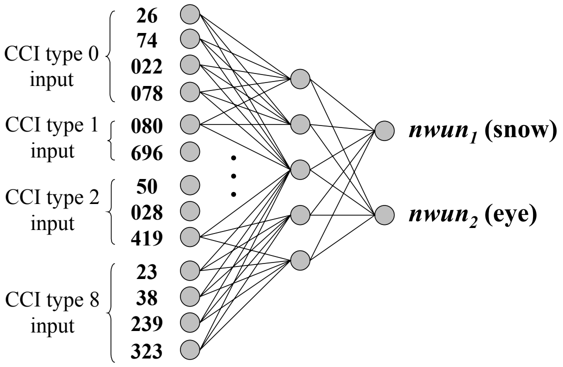

Figure 5. The Resulting Network for '*nwun*'

3.2 Network Learning

We selected 875 Korean homographs requring the WSD processing in a Korean-to-Japanese translation. Among the selected nouns, 736 nouns (about 84%) had two senses and the other 139 nouns had more than 3 senses. Using the extracted CCI, we constructed neural networks and trained network parameters for each homograph. The training patterns were also extracted from the previously constructed sense-tagged corpus.

The average number of input features (i.e. input nodes) of the constructed networks was approximately 54.1 and the average number of senses (i.e. output nodes) was about 2.19. In the case of a 2-layer network, the total number of parameters (synaptic weights) needed to be trained is about 118 (54.1×2.19) for each homograph. This means that we merely need storage for 118 floating point numbers (for synaptic weights) and 54 integers (for input features) for each homograph, which is a reasonable size to be used in real applications.

4 Word Sense Disambiguation

Our WSD approach is a hybrid method, which combines the advantage of corpus-based and knowledge-based methods. Figure 6 shows our overall WSD algorithm. For a given homograph, sense disambiguation is performed as follows. First, we search a collocation dictionary. The Korean-to-Japanese translation system COBALT-K/J has an MWTU (Multi-Word

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

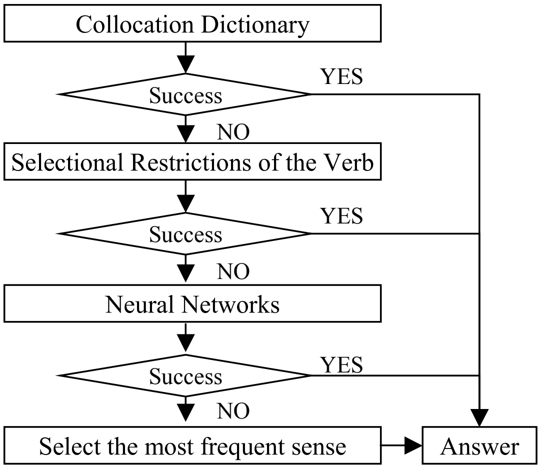

Figure 6. The Proposed WSD Algorithm

Translation Units) dictionary, which contains idioms, compound words, collocations, etc. If a collocation of the target word exists in the MWTU dictionary, we simply determine the sense of the target word to the sense found in the dictionary. This method is based on the idea of 'one sense per collocation'. Next, we verify the selectional restriction of the verb described in the dictionary. If we cannot find any matched patterns for selectional restrictions, we apply the neural network approach. WSD in the neural network stage is performed in the following 3 steps.

Step 1. Extract CCI from the context of the target word. The window size of the context is a single sentence. Consider, for example, the sentence in Figure 7 which has the meaning of "Seeing her eyes filled with tears, …". The target word is the homograph '*nwun*'. We extract its CCI from the sentence by partial parsing and pattern scanning. In Figure 7, the words '*nwun*' and '*kunye*(her)' with the concept code 503 have the relation of <noun + *uy* + noun>, which corresponds to '*CCI type 2*' in Table 1. There is no syntactic relation between the words *'nwun'* and '*nwunmul*(tears)' with the concept code 078, so we assign '*CCI type 0'* to the concept code 078.

Similarly, we can obtain all pairs of CCI types and their concept codes appearing in the context. All the extracted <*CCI-type*: *concept codes*> pairs are as follows: {<*type 0*: *078,274*>, <*type 2*: *503*>, <*type 8*: *331*>}.

Step 2. Obtain the input pattern for the

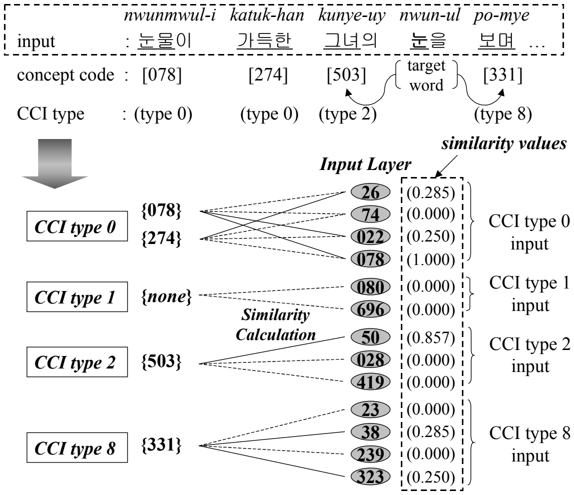

Figure 7. Construction of Input Pattern by Using Concept Similarity Calculation

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

Csim(C_i, P_j) in Equation 1 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. That is, if C_i is a descendant of P_j, we set *weight* to 1. Otherwise we set *weight* to 0.5.

$$Csim(C_i, P_j) = \frac{2 \times level(MSCA(C_i, P_j))}{level(C_i) + level(P_j)} \times weight \quad (1)$$

The similarity values between the target

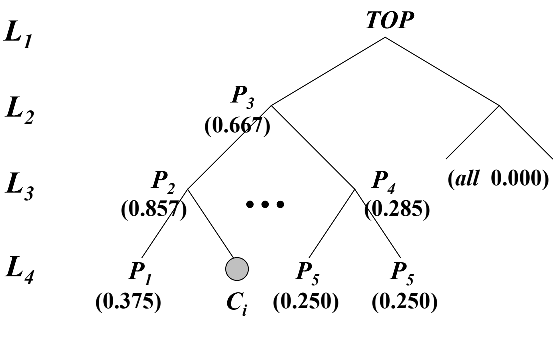

Figure 8. Concept Similarity on the Kadokawa Thesaurus Hierarchy

concept C_i and each P_j on the Kadokawa thesaurus hierarchy are shown in Figure 8. These similarity values are computed using Equation 1. For example, in '*CCI-type 0*' part calculation, the relation between the concept codes 274 and 26 corresponds to the relation between C_i and P_4 in Figure 8. So we assign the similarity 0.285 to the input node labeled by 26. As another example, the concept codes 503 and 50 have a relation between C_i and P_2 and we obtain the similarity 0.857. If more than two concept codes exist in one CCI-type, such as <*CCI-type 0*: *078, 274*>, the maximum similarity value among them is assigned to the input node, as in Equation 2.

$$InputVal(C_i) = \max_{P_i}(Csim(C_i, P_j)) \qquad (2)$$

In Equation 2, 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 as C_i.

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

Step 3. Feed the obtained input pattern to the neural network and compute activation strengths for each output node. Next, select the sense of the node that has a larger activation value than all other output node. If the activation strength is lower than the threshold, it will be discarded and the network will not make any decisions. This process is represented in Figure 9.

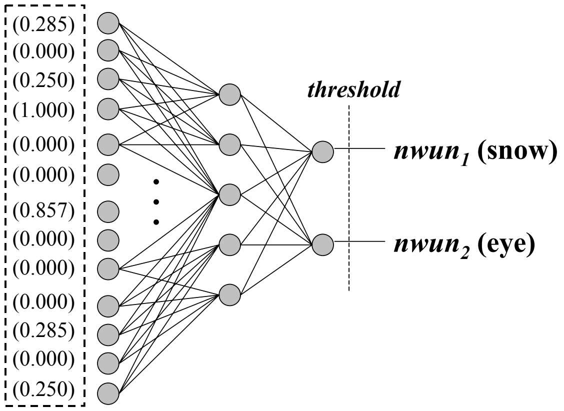

Figure 9. Sense Disambiguation for '*nwun*'

5 Experimental Evaluation

For an experimental evaluation, 10 ambiguous Korean nouns were selected, along with a total of 500 test sentences in which one homograph appears. In order to follow the ambiguity distribution described in Section 3.2, we set the number of test nouns with two senses to 8 (80%). The test sentences were randomly selected from the KIBS (Korean Information Base System) corpus.

The experimental results are shown in Table 3, where result **A** is the case when the most frequent sense was taken as the answer. To compare it with our approach (result **C**), we also performed the experiment using Li's method (result **B**). For sense disambiguation, Li's method features which are similar to our method. However, unlike our method, which combines all features by using neural networks, Li considers only one clue at each decision step. As shown in the table, our approach exceeded Li's

Table 3. Comparison of WSD Results

Word	Sense	No	Precision (%)		
			(A)	(B)	(C)
pwuca	father & child	33	66	64	72
	rich man	17			
kancang	liver	37	74	84	74
	soy source	13			
kasa	housework	39	78	68	82
	words of song	11			
kwutwu	shoes	45	90	70	92
	word of mouth	5			
nwun	eye	42	84	80	86
	snow	8			
yongki	container	41	82	72	88
	courage	9			
uysa	doctor	27	54	80	84
	intention	23			
cikwu	district	27	54	84	92
	the earth	23			
censin	whole body	39	78	84	80
	one's past	6			
	telegraph	5			
cenlyek	one's best	27	54	50	72
	military strength	13			
	electric power	7			
	past record	3			
Average Precision			**71.4**	**73.6**	**82.2**

※ (A) : Baseline (B) : Li's method
(C) : Proposed method (using a 2-layer NN)

Table 4. Average Precision and Coverage for Each Stage of thePproposed Method

<Case 1 : 2-layer NN>

	COL	**VSR**	**NN**	**MFS**
Avg. Prec	100.0%	91.2%	**86.3%**	56.1%
Avg. Cov	3.6%	6.8%	**73.2%**	16.4%

<Case 2 : 3-layer NN>

	COL	**VSR**	**NN**	**MFS**
Avg. Prec	100.0%	91.2%	**87.1%**	56.0%
Avg. Cov	3.6%	6.8%	**72.5%**	17.1%

in most of the results except '*kancang*' and '*censin*'. This result shows that word sense disambiguation can be improved by combining several clues together (e.g. neural networks) rather than using them independently (e.g. Li's method).

The performance for each stage of the proposed method is shown in Table 4. Symbols COL, VSR, NN and MFS in the table indicate 4 stages of our method in Figure 6, respectively. In the NN stage, the 3-layer model did not show a performance superior to the 2-layer model because of the lack of training samples. Since the 2-layer model has fewer parameters to be trained, it is more efficient to generalize for limited training corpora than the 3-layer model.

Conclusion

To resolve sense ambiguities in Korean-to-Japanese MT, this paper has proposed a practical word sense disambiguation method using neural networks. Unlike most previous approaches based on neural networks, we reduce the number of features for the network to a practical size by using concept codes rather than lexical words. In an experimental evaluation, the proposed WSD model using a 2-layer network achieved an average precision of 82.2% with an improvement over Li's method by 8.6%. This result is very promising for real world MT systems.

We plan further research to improve precision and to expand our method for verb homograph disambiguation.

Acknowledgements

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center(AITrc).

References

Gallant S. (1991) *A Practical Approach for Representing Context and for Performing Word Sense Disambiguation Using Neural Networks*. Neural Computation, 3/3, pp. 293-309

Leacock C., Twell G. and Voorhees E. (1993) *Corpus-based Statistical Sense Resolution.* In Proceedings of the ARPA Human Language Technology Workshop, San Francisco, Morgan Kaufman, pp. 260-265

Li H. F., Heo N. W., Moon K. H., Lee J. H. and Lee G. B. (2000) *Lexical Transfer Ambiguity Resolution Using Automatically-Extracted Concept Co-occurrence Information.* International Journal of Computer Processing of Oriental Languages, 13/1, pp. 53-68

McRoy S. (1992) *Using Multiple Knowledge Sources for Word Sense Discrimination*. Computational Linguistics, 18/1, pp. 1-30

Mooney R. (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, H. T. and Zelle J. (1997) *Corpus-Based Approaches to Semantic Interpretation in Natural Language Processing*. AI Magazine, 18/4, pp. 45-64

Ohno S. and Hamanishi M. (1981) *New Synonym Dictionary*. Kadokawa Shoten, Tokyo

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

Waltz D. L. and Pollack J. (1985) *Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation.* Cognitive Science, 9, pp. 51-74