JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 11, No. 1, March 2017, pp. 32-38

Word Sense Disambiguation Using Embedded Word Space

Myung Yun Kang
Department of Business Data Convergence, Chungbuk National University, Cheongju, Korea
myungyun@cbnu.ac.kr

Bogyum Kim and Jae Sung Lee*
Department of Computer Science, Chungbuk National University, Cheongju, Korea
bogyum@cbnu.ac.kr, jasonlee@cbnu.ac.kr

Abstract
Determining the correct word sense among ambiguous senses is essential for semantic analysis. One of the models for word sense disambiguation is the word space model which is very simple in the structure and effective. However, when the context word vectors in the word space model are merged into sense vectors in a sense inventory, they become typically very large but still suffer from the lexical scarcity. In this paper, we propose a word sense disambiguation method using word embedding that makes the sense inventory vectors compact and efficient due to its additive compositionality. Results of experiments with a Korean sense-tagged corpus show that our method is very effective.

Category: Human computing

Keywords: Word sense disambiguation; Word embedding; Word space; Semantic analysis

I. INTRODUCTION

Determining the correct meaning of a word, especially a homograph, is very important for the performance of natural language processing systems, such as information retrieval systems, machine translation systems, question answering systems, and others. The meaning of a homograph is often determined by its contextual words. For example, 'bank' means both a monetary institute and a slope of land bordering a river. Based on the contexts of the following sentences, the respective meanings of the word can be readily identified:

I go to the *bank* to deposit money.
I walk along the *bank* of the river.

In most word sense disambiguation (WSD) systems, the neighboring words and part-of-speech tags are used as clues and are represented as elements in a vector [1]. A word vector to be disambiguated is typically called a 'context vector,' while the target sense in an inventory is the 'sense vector.' In other words, WSD is the task of finding the most similar sense vector in a word sense inventory for a given context vector.

WSD is performed in knowledge-based, supervised, and unsupervised approaches [2]. Knowledge-based approaches use for sense vectors lexical resources, such as dictionaries and thesauruses [3-5]. However, knowledge-based approaches suffer from a data sparseness problem mainly because the knowledge sources, such as dictionaries and thesauruses, do not provide enough lexical entries for matching with a context vector.

Supervised approaches use a sense-tagged corpus for

Open Access http://dx.doi.org/10.5626/JCSE.2017.11.1.32 http://jcse.kiise.org

Received 06 January 2017; **Revised** 12 March 2017; **Accepted** 15 March 2017
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

training the systems. They are predominantly based on machine learning methods. The sense vectors are directly gathered from the corpus and are represented in vector space [6-8]. Supervised approaches provide more lexical information from the training corpus; however, the corpus is expensive and is limited to small size.

Unsupervised approaches use a raw corpus and induce senses from the corpus for labeling senses [9]. Unsupervised approaches partially solve the cost problem of the supervised approaches by automatically tagging the ambiguous words with 'derived senses' from the raw corpus itself.

As more corpus and language resources are utilized for context representation, the sizes of sense vectors grow rapidly, which demands a size reduction and effective representation of vectors. Dimension reduction is useful not only for size reduction of a sense vector inventory, but also for mitigating the data sparseness problem.

In this paper, we propose a method for word sense disambiguation that represents context and sense vectors with 'word vectors' produced by a deep neural net program, Word2Vec (W2V) [10]. We use a sense-tagged corpus to create sense vectors and then reduce the dimension of vectors by using a continuous bag of word (CBOW) architecture in W2V. Our proposed model is "an embedded word space model" which expands the previous researches of word space model [9, 11, 12] to incorporate embedded word vectors. In this model, all the word types are represented in the same word space and WSD is done by simple calculation of vector similarity between homographs.

In Section II, we briefly review the related works. In Section III, we overview the word space model and its modified model. Our proposed method is described in Section IV, and its model features and combinations are presented in Section V. Results of experiments conducted with a Korean sense-tagged corpus are provided in Section VI, and our conclusions are discussed in Section VII.

II. RELATED WORKS

Schutze [9] used singular value decomposition (SVD) for reducing the dimension of context vectors and obtaining more effective sense induction from the unlabeled corpus. Gliozoo et al. [13] used the latent semantic analysis (LSA) method to induce from unlabeled corpus a domain matrix that maps external terms to the terms in the supervised learning process. This process enables the WSD system to process the terms in different domains. Cai et al. [14] and Li et al. [15] constructed a topic model using latent Dirichlet allocation (LDA) to mitigate the data scarcity problem by mapping words to topics in a sense selection process.

Using a deep neural net is known to be more effective for semantic relation testing than principal component analysis or LSA and is computationally less expensive than LDA [16, 17]. Recently word embedding is used for WSD with various word weights [18] where word vectors are used for part of features of support vector machine (SVM) learning. It showed that using word vector features is effective for the WSD. However, the multiclass SVM needs to be trained for each word type, which produces as many models as the number of word types.

III. WORD SPACE MODEL

The word space model used for WSD [9] is an application of the vector space model used in information retrieval [19]. In the model, all words are represented in vectors, and the similarity between words is measured with a vector distance. A sense vector is represented with context words, w_j, appearing around word sense s_i, as shown in Eq. (1), where V is the size of the vocabulary, and $f(w_j)$ is the weight function of word w_j.

$$vec(s_i) = (f(w_1), f(w_2), \ldots f(w_V)) \quad (1)$$

Additionally, a context vector of a query word, qw, is represented in the same way.

$$vec(qw) = (f(qw_1), f(qw_2), \ldots f(qw_V)) \quad (2)$$

WSD finds the most similar word sense vector among the word vectors in the sense inventory belonging to the surface form of qw.

$$sense(qw) = \underset{k}{\operatorname{argmax}} \quad sim(vec(qw), vec(s_k)) \quad (3)$$

Previous studies show that using prior probability is effective [11, 20]. The Bayesian method used in [20] can be modified to use prior probability and cosine similarity by replacing conditional probability with the cosine similarity plus one to have positive values [12]. The method is represented in Eq. (4). In this paper, we further derive the equation to use the adjusting factor alpha to be balanced with prior probability as shown in Eq. (5). Because the *argmax* operator does not need normalization of arguments, powering the first argument with positive integer values of alpha is valid.

$$\begin{aligned}
&\underset{k}{\operatorname{argmax}} \;\; pr(s_k|qw) \\
&= \underset{k}{\operatorname{argmax}} \;\; pr(qw|s_k) \cdot pr_{prior}(s_k) \\
&\approx \underset{k}{\operatorname{argmax}} \;\; (1+\cos(vec(qw), vec(s_k))) \cdot pr_{prior}(s_k) \quad (4) \\
&\approx \underset{k}{\operatorname{argmax}} \;\; (1+\cos(vec(qw), vec(s_k)))^{\alpha} \cdot pr_{prior}(s_k) \quad (5)
\end{aligned}$$

IV. EMBEDDED WORD SPACE MODEL

The W2V program produces 'induced word vectors' by learning word sequences. The induced word vectors have compositionality, which makes a vector addition possible. For example, vec('king') – vec('man') + vec('woman') = vec('queen') [16]. Assuming that the vector operation is valid for the semantic calculation, we use the induced word vectors for the dimension reduction.

First, each word sense is represented with context words by the same method as the word space model. Then, the W2V program is trained with word sequences (these are not word sense sequences). The resulting vectors are used for the addition of each corresponding context word to make the sense vectors compact. The dimension of the induced word vector is typically much smaller than the vocabulary size, and the addition results in dimension reduction. This process is shown in Eq. (6). The weight is multiplied for each word vector of context word w_j to reflect the importance of the specific word.

$$vec(s_i) = \sum_j w2v(w_j)f(w_j) \tag{6}$$

The query word vector can be represented in the same way by using the query context words, qw_j, and by adding their corresponding induced vectors to make a compact vector:

$$vec(qw) = \sum_j w2v(qw_j)f(qw_j) \tag{7}$$

If a word appears less than a threshold, the word is usually replaced with an *unknown* marker in W2V learning [10]. Therefore, some infrequent words in contexts are replaced with *unknown* word vectors.

V. FEATURES AND COMBINED MODELS

We may have various combinations of the two base models with various parameters. For parameters, we consider the word weight function, default sense selection, and context word selection basis as follows.

Word weight function: The weighted function can be defined in various ways by combining the frequency, word distance, chi-square, and inverse document frequency (IDF). We chose Eq. (8), which showed the best performance among the various combinations in preliminary testing. As the dimension of sense vectors is usually very large, dimension reduction is thus performed in choosing context words. First, we limit the context words to content words (noun, verb, adjective). Second, we select the high frequency words close to the target word within a fixed window size (in this paper, five). In the second case, two factors are multiplied, as shown in Eq. (8), to be used for the word weight and dimension cut (The word distance used in this paper is similar to the fractional decay in [18], where exponential decay showed better performance. However, we adopt our distance calculation formula shown in (8) for the comparison with previous works [11, 20]).

$$f(w_j) = \sum_k freq(w_{jk})distw(w_{jk}) \tag{8}$$

$$distw(w) = WindSize - OffsetFromSi(w) + 1$$

Default sense: In all models, the sizes of the inventory words are not large enough to cover all the context words. When none of the sense vector candidates match with a context vector, we select the most frequent sense (MFS) as a default sense to alleviate the data sparseness problem.

Local and global word selection: The context word frequency can be measured either locally for each homograph basis or globally for all the homograph bases. In local selection, each homograph has different vector space, while in global selection, each homograph has the same vector space. Consequently, local selection is implemented with a more complex data structure.

Combining all the features with a word space model (WS) and an embedded word space model (WE), we choose four major models as follows:

1. Local WS: local word selection, using frequency * distance weight function, WS model with default MFS.
2. Global WS: global word selection, using frequency * distance weight function, WS model with default MFS.
3. Global WE: global word selection, using frequency * distance weight function, WE model with default MFS.
4. MFS: baseline model using MFS only. This is equal to the model that uses only prior probability by setting alpha to zero in Eq. (5).

Note that the local WE model is meaningless because WE globally functions.

VI. EXPERIMENT

A. Experimental Setup

We used the Korean Sejong sense-tagged corpus [21] for our experiments in which the homographs were tagged with sense number labels. Table 1 shows the characteristics of the corpus.

Approximately 8% of homographs were missing labels in the corpus. We thus ignored them in the evaluation. Both the window size of the W2V training and that of the context vectors were five. In addition, the input words were limited to the content words: nouns, verbs, and

Table 1. Statistics of the Sejong sense-tagged corpus

No. of sentences	No. of word phrases	No. of sense-tagged words	No. of homograph types	Average no. of senses per homograph
832,650	9,524,183	3,892,113	17,078	1.6

Table 2. Best performances of the three proposed models and baseline model

Model	Dimension						
	25	50	100	200	300	400	500
Global WE	91.28 (95.58)	91.33 (96.01)	91.40 (96.02)	91.40 (96.03)	91.38 (96.03)	**91.41** **(96.03)**	91.39 (96.04)
Local WS	89.62 (94.95)	90.02 (95.30)	90.38 (95.64)	90.67 (95.93)	90.76 (96.07)	90.81 (96.15)	**90.84** **(96.20)**
Global WS	87.99 (94.08)	88.13 (94.23)	88.30 (94.43)	88.58 (94.69)	88.78 (94.87)	88.95 (95.01)	**89.06** **(95.12)**
MFS	87.87 (94.03)	87.87 (94.03)	87.87 (94.03)	87.87 (94.03)	87.87 (94.03)	87.87 (94.03)	87.87 (94.03)

Macro (micro) average precision in %, 15 negative sampling, 5 min cut.
WE: embedded word space model; WS: word space model; MFS: most frequent sense.

adjectives. We tested the CBOW W2V architecture with a few parameter settings and chose the best one. The selected option for the architecture was negative sampling with 15 words. We additionally used a minimal frequency cut of five and handled unknown words.

We ran 10-fold cross-validation tests with 25, 50, 100, 200, 300, 400, and 500 respective dimensions, combined with alpha values ranging from one to ten. We chose the best performance in all tested dimensions and variable values of alpha for each of the three models: local WS, global WS, and global WE (CBOW).

We use two measures, macro average precision and micro average precision (In this experiment, precision is the same as recall because the test system produces one output for each homograph in test data). Macro average is calculated as the average of each homograph type's average precision, while micro average is as the average precision of all homograph instances. Because micro average is dominated by large category, macro average is preferred for the quality measure of classification across all categories, in this case, homograph types [22].

B. Results

Table 2 shows the precision of the baseline and the three models in all tested dimensions when the alpha is set to the best performing values: the value ranges from six to eight in global WE, two in the local WS, and one in the global WS. The baseline is the model of most frequent sense selection method. Its performance is relatively high, which is explained by the low average number of senses per homograph, as shown in Table 1.

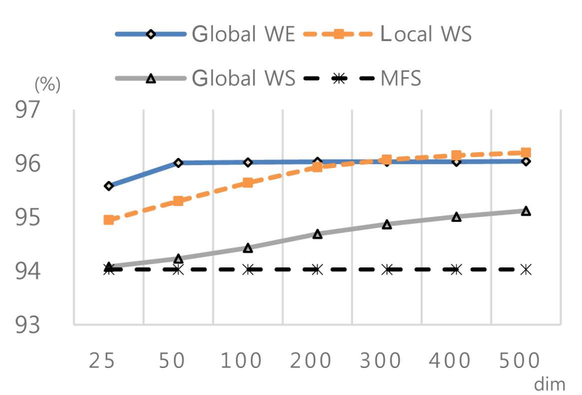

Fig. 1. Micro average precision of the models in various dimensions. WE: embedded word space model, WS: word space model, MFS: most frequent sense.

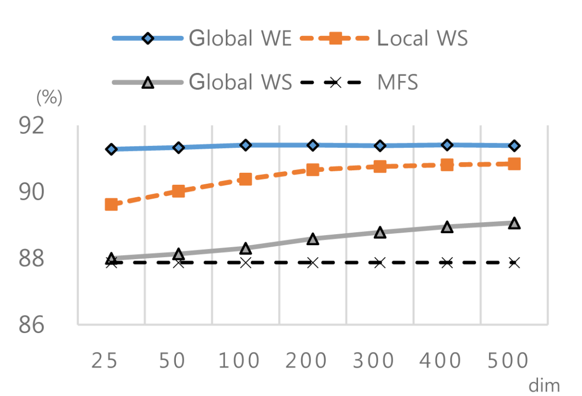

Fig. 2. Macro average precision of the models in various dimensions. WE: embedded word space model, WS: word space model, MFS: most frequent sense.

The micro average precision (represented within parenthesis in Table 2) of all the models is higher than the macro average, which means the data are skewed and the frequent homographs are better in performance than infrequent ones. In this measure, global WE is the highest in the low dimension (from 25 to 200); however, it is the second highest in the high dimension (from 300 to 500), as shown in Table 2 and Fig. 1. Nevertheless, in the macro average, the general performance of the global WE is highest in all dimensions, as shown in Table 2 and Fig. 2.

The adjusting factor, alpha, plays an important role in improving the performance of global WE. The alpha value adjusts the distribution difference of the similarity of word vectors and the prior probability. For example, without adjusting alpha, the incorrect sense (지면__03/NNG) is selected, as shown in case 1 in Table 3. However, with a properly adjusted alpha, the correct sense (지면__01/NNG) is selected, as shown in case 2.

Table 4 shows the comparison with the results of other Korean word sense disambiguation studies. The proposed model is competitive, even though the testing environments are different.

We selected our corresponding test result to the 4 homographs used in Lee et al. [23] as shown in Table 5. The average performance of those words is slightly lower than Lee et al.'s result. We conjecture two reasons for the lower performance: (1) our test set has more senses per homograph (average 4.5 vs. 4.0) and (2) our model is optimized to much larger number of homographs (about

Table 3. Examples of balance factor adjusting for sense selection (answer is *지면__01/NNG and cosa = $(\cos(q,s)+1)^{\alpha}$)

	sense	cosa	prior	cosa × prior
Case 1 (without adjusting factor)	*지면__01/NNG	0.728	0.086	0.062
	지면__02/NNG	0.496	0.005	0.003
	지면__03/NNG	0.443	0.765	**0.339**
	지면__04/NNG	0.429	0.144	0.062
Case 2 (with adjusting factor alpha = 6)	*지면__01/NNG	0.148	0.086	**0.013**
	지면__02/NNG	0.015	0.005	0.000
	지면__03/NNG	0.008	0.765	0.006
	지면__04/NNG	0.006	0.144	0.001

Test data: "백스윙이 톱에 이를 때 발바닥을 **지면**에 붙이면" (If you put your feet on the ground when you reach the top of the backswing).
Senses: 지면__01 (ground), 지면__02 (acquaintance), 지면__03 (paper or in printed page), 지면__04 (in a magazine).

Table 4. Comparison with other Korean research results

Model	Resources and method	Testing data	Precision (%) (macro avg.)
Lee et al. [23]	Sense tagged corpus, entropy	4 homographs	84.6
Kim et al. [20]	Dictionary, co-occurrence	46 homographs	74.1
Heo et al. [24]	Dictionary, raw corpus, sense tagged multi-words, mutual information	About 200 homographs	88.8
Proposed	Sense tagged corpus, embedded word space	About 17,000 homographs	91.4

Table 5. Selective comparison with Lee et al.'s result

Homographs	Lee et al. [23]		Our model	
	Precision (%)	No. of senses	Precision (%)	No. of senses
배/NNG	82.20	4	72.19	9
전자/NNG	92.20	2	91.66	4
감다/VV	81.70	3	74.30	3
열리다/VV	82.40	7	98.08	2
Average	84.63	4.0	84.06	4.5

Senses: 배/NNG (stomach, ship, pear, …), 전자/NNG (electron, former, ...), 감다/VV (close, wind, wash), 열리다/VV (open, bear fruit, …).

17,000 vs. 4). Considering these facts, our model is still competitive to the Lee et al.'s model.

Shin [25] proposed an integrated method for simultaneous morphological analysis (MA), part-of-speech (POS) tagging, and WSD using a pre-analyzed word phrase dictionary. In order to solve data sparseness problem in dictionary-based approach, he used partial matching with some heuristic rules and probabilistic approaches. He estimated micro average precision of the WSD performance by calculating the ratio of the number of word phrases, which are tagged correctly with POS and senses, to the two different denominators: (1) the number of all word phrases including homographs and (2) the number of word phrases including homographs and correctly analyzed MA/POS result. The first one is 95.8%, which may be degraded by the propagation error from MA and POS tagging stages, is lower than our best micro average result of global WE model, 96.0%. The second one is 98.5%, which we conjecture could be exaggerated by excluding error-prone word phrases such as low frequency word phrases, is higher than our result (Note that our test set includes all the word phrases that are manually POS tagged). Therefore, we cannot judge easily which one is better, but we conjecture our method is comparable to Shin's method, too.

In some cases, the distribution of senses is skewed and sense similarity does not contribute to the performance much. For example, a homograph, 전/NNG, has 15 senses in a training set and the sense 전_08/NNG (before) is dominant by 98.8%. Therefore, it is very hard to discriminate other senses from 전_08/NNG unless we set the alpha to a very big number in Eq. (5). For further improvement, it may be needed to set the different balancing factor, alpha, for each homograph depending on the sense distribution.

As we focused on Korean word sense disambiguation task, we tested our model only for Korean data set. As our proposed method is language independent, we believe that it can be applied to other languages, which is left for future research.

VII. CONCLUSION

Word sense disambiguation systems use a considerable amount of resources to include appropriate context information. Moreover, the representation of the context vector is important for the system accuracy. In this paper, we have proposed a method for word sense disambiguation using word embedding, which makes the representation of context word vectors compact and accurate for sense selection. Moreover, the model is simple so that all the word types can be disambiguated within one embedded word space. The results of experiments with a Korean sense-tagged corpus show that the proposed method using word embedding is effective.

ACKNOWLEDGMENTS

This work was supported by ICT R&D program of MSIP/IITP (No. R0101-15-0062; Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services) and also supported by the research grant of Chungbuk National University in 2014.

REFERENCES

1. A. Kilgarriff and J. Rosenzweig, "Framework and results for English SENSEVAL," *Computers and the Humanities*, vol. 34, no. 1-2, pp. 15-48, 2000.
2. D. McCarthy, "Word sense disambiguation: an overview," *Language and Linguistics compass*, vol. 3, no. 2, pp. 537-558, 2009.
3. M. Lesk, "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, Canada, 1986, pp. 24–26.
4. D. E. Walker, "Knowledge resource tools for accessing large text files," in *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Hamilton, NY, 1985, pp. 335-347.
5. D. Yarowsky, "Word-sense disambiguation using statistical models of Roget's categories trained on large corpora," in *Proceedings of the 14th Conference on Computational Linguistics (COLING)*, Nantes, France, 1992, pp. 454-460.
6. Y. Wilks, D. Fass, C. M. Guo, J. E. McDonald, T. Plate, and B. M. Slator, "Providing machine tractable dictionary tools," *Machine Translation*, vol. 5, no. 2, pp. 99-154, 1990.
7. W. A. Gale, K. W. Church, and D. Yarowsky, "A method for disambiguating word senses in a large corpus," *Computers and the Humanities*, vol. 26, no. 5-6, pp. 415-439, 1992.
8. P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer, "Word-sense disambiguation using statistical methods," in *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, Berkeley, CA, 1991, pp. 264-270.
9. H. Schutze, "Automatic word sense discrimination," *Computational Linguistics*, vol. 24, no. 1, pp. 97-123, 1998.
10. Word2Vec, https://code.google.com/p/word2vec/.
11. Y. M. Park and J. S. Lee, "Word sense disambiguation using Korean word space model," *Journal of the Korea Contents Association*, vol. 12, no. 6, pp. 41-47, 2012.
12. M. Y. Kang, B. Kim, and J. S. Lee, "Word sense disambiguation using Word2Vec," in *Proceedings of the 26th Annual Conference on Human and Cognitive Language Technology*, Busan, 2016.
13. A. Gliozzo, C. Giuliano, and C. Strapparava, "Domain kernels for word sense disambiguation," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, MI, 2005, pp. 403-410.
14. J. F. Cai, W. S. Lee, and Y. W. Teh, "NUS-ML: improving word sense disambiguation using topic features," in *Proceedings of the 4th International Workshop on Semantic Evalua-*

tions, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 249-252.
15. L. Li, B. Roth, and C. Sporleder, "Topic models for word sense disambiguation and token-based idiom detection," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 1138-1147.
16. T. Mikolov, W. T. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, Atlanta, GA, 2013, pp. 746-751.
17. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of ICLR workshop*, 2013.
18. I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Embeddings for word sense disambiguation: an evaluation study," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, 2016, pp. 897-907.
19. G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613-620, 1975.
20. J. S. Kim, H. S. Choe, and C. Y. Ock, "A Korean homonym disambiguation model based on statistics using weights," *Journal of KIISE: Software and Applications*, vol. 30, no. 11, pp. 1112-1123, 2003.
21. The National Institute of the Korean Language, "21st Century Sejong Project Final Result, revised edition," 2011.
22. C. D. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.
23. H. Lee, D. H. Baek, and H. C. Rim, "Word sense disambiguation using classification information," *Journal of KIISE: Software and Applications*, vol. 24, no. 7, pp. 779-789, 1997.
24. J. Heo, H. C. Seo, and M. G. Jang, "Homonym disambiguation based on mutual information and sense-tagged compound noun dictionary," *Journal of KIISE: Software and Applications*, vol. 33, no. 12, pp. 1073-1089, 2006.
25. J. C, Shin, "Korean morphological analysis based on pre-analyzed partial word-phrase dictionary and part-of-speech/homograph tagging based on syllable-morpheme transition probability," Ph.D. dissertation, Ulsan University, 2013.

Myung Yun Kang

Myung Yun Kang received the B.S. degree in computer education and M.S. degree in business intelligence from Chungbuk National University in 2015 and 2017, respectively. Currently, he works as a software engineer for Daumsoft in Korea. His research interests include natural language processing, machine learning, text mining, and deep learning.

Bogyum Kim

Bogyum Kim received the B.S. and M.S. degrees in computer education, the Ph.D. degree in digital informatics and convergence from Chungbuk National University in 2007, 2011, and 2016, respectively. Currently, he works as a software engineer for Wemakeprice Inc. in Korea. His research interests include natural language processing, text mining, and information retrieval.

Jae Sung Lee

Jae Sung Lee received his B.S. degree in computer engineering from Seoul National University, Seoul, Korea in 1983, and M.S. and Ph.D. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea in 1985 and 1999, respectively. He was a visiting scholar in University of Arizona from July 2005 to June 2006. He is currently a professor in the Department of Computer Science, Chungbuk National University, Korea. He worked for Qnix Computer Corp. and Microsoft Corp., where he developed Korean versions and double-byte code versions of system software such as Korean BIOS (Basic Input Output System), MS-DOS, and MS-Windows. He also worked as a team leader for Electronics and Telecommunications Research Institute (ETRI). His research interests include natural language processing, information retrieval, machine learning, Korean text processing, and terminology engineering.