Joint Model of Korean Part-of-Speech Tagging and Dependency Parsing with Partial Tagged Corpus

Youngmin Park and Jungyun Seo

***Abstract*—Most recent studies on part-of-speech (POS) tagging and dependency parsing employ a pipelined model design. However, pipelined structures may decrease performance on account of error propagations. Furthermore, syntactic information is required to improve POS tagging performance. In this paper, we propose a joint model of POS tagging and dependency parsing for the Korean language. Our joint model analyzes the maximum score dependency tree with POS tagging using the graph-based CKY parsing method. We present an effective application method for an additional POS tagged corpus and evaluate the method for POS tagging and dependency parsing. The results show that our model improves accuracy by approximately 2.3% and 1.7% more than a pipelined structure using a hidden Markov model and graph-based dependency parsing, respectively.**

***Index Terms*—Part-of-speech tagging, dependency parsing, natural language processing, joint model.**

I. Introduction

In natural language processing tasks, part-of-speech (POS) tagging is the process of marking or tagging each word in a text as corresponding to a particular part of speech (nouns, verbs, adjectives, etc.). Dependency parsing is used to analyze syntactic structures as dependency relations among words in a sentence. POS tagging and dependency parsing are widely used in text-based technology, including information extraction [1], machine translation [2], opinion mining [3], information retrieval [4], and so on. POS tagging and dependency parsing are typically designed with a pipelined structure that analyzes POS tags before dependency parsing. POS tag features are very effective in dependency parsing; moreover, most POS taggers provide better performance than dependency parsers. However, the pipelined structure may decrease performance on account of error propagations in the POS tagging step. Furthermore, POS tagging often requires syntactic information, such as dependency relations.

In conventional Korean natural language processing, morphological analysis, POS tagging, and dependency parsing are designed with a pipelined method, as shown in Fig. 1. A first component morphological analyzer generates possible candidates of morphemes and POS tags for each eojeol, which is a Korean spacing unit similar to an English word. Recent studies in the morphological analyzing step have shown recalls of approximately 99% [5]. In addition, a second component POS tagger decides the most acceptable candidate for each eojeol; it has shown an accuracy of approximately 95% [6], [7]. The most recent component dependency parser to analyze dependency relations between eojeols has demonstrated an accuracy of approximately 88% [8]. However, that performance was evaluated on the assumption that the precomponent provides complete output. If each component is evaluated from a raw sentence by the pipelined method, dependency parsing accuracy decreases to approximately 83%.

In this paper, we propose a joint model of POS tagging and dependency parsing for the Korean language. In addition, we modify the learning algorithm to apply only the POS-tagged corpus.

II. Related Works

Most studies of POS tagging have been applied to the sequence labeling problem with probabilistic models, such as hidden Markov models (HMMs) and conditional random fields (CRFs) [6], [9]. Typical studies of dependency parsing have classified it as one of two methodologies: graph-based or transition-based. Graph-based dependency parsing generates a tree that has the globally highest score [10], [11]. Transition-based dependency parsing analyzes transition sequences by classifying the highest scoring transition for a given state [12], [13]. The parser then generates a tree constructed by transition sequences. The typical graph-based method tends to provide better performance than the transition-based approach for long sentence length on account of its global search method. However, the transition-based approach tends to give better performance for short sentence length on account of its greedy method. Consequently, the dependency parser’s performance depends on the language to which it is applied [14].

Recently, joint models have been widely used to improve performance by integrating two or more tasks. Li *et al.* [15] applied a joint model to Chinese POS tagging and dependency parsing using a graph-based model and pruning method. This model showed better accuracy than a state-of-the-art parsing method by approximately 1.5%. Bohnet and Nivre [16] proposed another strategy that applies a transition-based model, which is a popular method for dependency parsing tasks. This model globally analyzes transition sequences and applies a beam search to reduce the computational cost; it then generates a non-projective tree. Based on evaluation results, it showed greater accuracy than a state-of-the-art method by approximately 1.2% on the Penn Chinese Treebank corpus.

Manuscript received February 8, 2015; revised April 8, 2015. This work was supported by ATC (Advanced Technology Center) Program – “Development of Conversational Q&A Search Framework Based on Linked Data: Project No. 10048448”.

The authors are with the Department of Computer Science and Engineering, Sogang University, Korea (e-mail: pymnlp@gmail.com, seojy@sogang.ac.kr).

DOI: 10.7763/IJKE.2015.V1.8

III. Baseline Model

For comparative experiments, we designed a baseline pipelined model of a morphological analyzer, POS tagger, and dependency parser.

A. Morphological Analyzer

In the first step of the proposed method, the morphological analyzer generates possible morphemes and POS tag candidates for a given input sentence x, as shown in Fig. 1. We used the UTAGGER module [5], [17] for this step to set our model equal to the baseline.

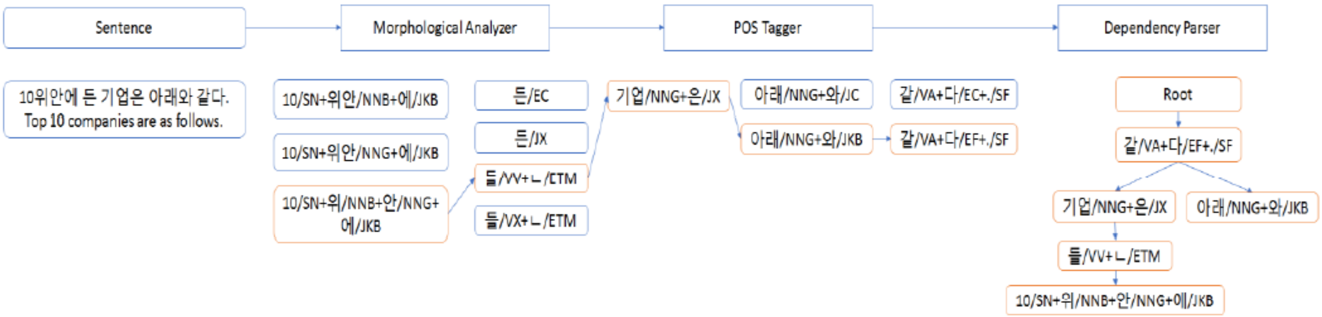


Fig. 1. Conventional pipelined structure for Korean natural language processing.

B. POS Tagger

We re-implemented the Korean POS tagger with HMMs to a baseline model that follows the work of [18]. Given an input sentence of $x = w_1, w_2, \ldots w_n$, the POS tagger statistically analyzes the optimal POS tag sequence of $t = t_1, t_2 \ldots t_n$. An optimal POS tag sequence, $\hat{t}$, is determined by the following formula:

$$\hat{t} = \operatorname*{argmax}_{t} P(T|W)$$

$$\cong \operatorname*{argmax}_{t} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

C. Dependency Parser

Recent studies in Korean dependency parsing tasks have shown that the graph-based approach to dependency parsing provides remarkable performance. Graph-based dependency parsing views the problem as finding the best-scoring tree among feasible trees. Therefore, an optimal dependency tree, $\hat{y}$, is determined by the following formula:

$$\hat{y} = \operatorname*{argmax}_{y} S(x, y)$$

where $S(x, y)$ refers to the score of dependency tree y for given input sentence *x*.

We re-implemented the graph-based Korean dependency parser to a baseline model that follows the work of [8]. This model computes the score of the dependency tree with an edge-factored model [19] that calculates the score of the tree by a summation of subtrees.

$$S(x, y) = \sum_{(i,j)\in y} s(i, j) = \sum_{(i,j)\in y} w \cdot f(i, j)$$

where$f(i, j)$ refers to the feature function of *eojeoli* and *j*. In addition, w is the corresponding weight vector trained by the averaged perceptron [20], [21] as a machine learning method.

IV. Joint Model

In this section, we describe our improved joint model of Korean POS tagging and dependency parsing. An overview diagram of the proposed model is shown in Fig. 2. The proposed model employs a fully tagged corpus (both POS and syntactic) as well as an only POS tagged corpus. Existing studies on such joint models have been limited to only using a fully tagged corpus [15], [16]. Thus, it has been difficult for the joint models to achieve a higher performance than the state-of-the-art POS tagging method on account of the small corpus problem. In fact, the Sejong Korean tagged corpus consists of an approximately 65,000 fully tagged and approximately 1,000,000 POS tagged corpus [22].

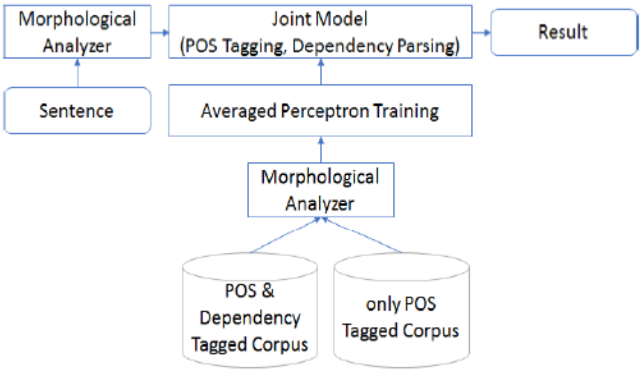

Morphological Analyzer
Joint Model
(POS Tagging, Dependency Parsing)
Result
Sentence
Averaged Perceptron Training
Morphological Analyzer
POS & Dependency Tagged Corpus
only POS Tagged Corpus

Fig. 2. Joint model of POS tagging and dependency parsing.

A. Parsing Algorithm

We resolve the joint model as a graph-based method that finds the best scoring tree consisting of a POS tag and dependency. The best scoring POS tag, $\hat{t}$, and the dependency tree, $\hat{y}$, are analyzed by the following formula:

$$(\hat{t}, \hat{y}) = \operatorname*{argmax}_{t,y} S(x, t, y)$$

The score function $S(x, t, y)$ can then be divided into a POS score part, S_{pos}, and a dependency score part, S_{dep}.

$$S(x, t, y) = S_{pos}(x, t) + S_{dep}(x, t, y)$$

Each score part is measured by the following formula:

$$S_{pos}(x, t) = \sum_{i\in t} s_{pos}(i) = \sum_{i\in t} w_{pos} \cdot f_{pos}(i)$$

$$S_{dep}(x,t,y) = \sum_{(i,j)\in y} s_{dep}(i,j) = \sum_{(i,j)\in y} w_{dep} \cdot f_{dep}(i,j)$$

where f_{pos} and f_{dep} denote the feature functions to be extracted from the POS and dependency relation. The weight vectors, which correspond to the feature functions w_{pos} and w_{dep} , are simultaneously updated in the training step.

To analyze the best scoring tree, we employ a dynamic programming method with a bottom-up CKY algorithm that applies the head-final and projectivity properties of the Korean language. Fig. 3 represents a graphical specification of the parsing algorithm. Triangles denote complete spanning trees; trapezoids represent incomplete spanning trees. Rectangles signify *eojeol* node sets; circles within rectangles denote nodes that are POS candidates.

The pseudo codes of the parsing algorithms are shown in Fig. 4. The function *candidate_size_of(i)* returns the number of POS candidates of *eojeoli*. Lines 3 to 5 are used to compute the scores of the terminal nodes that correspond with circles in Fig. 4. Lines 6 to 10 are used to explore feasible trees with a bottom-up process. It is possible to identify the optimal maximum spanning tree by back-referencing the result chart. Then, the parsing algorithm requires the time complexity of $O(n^3m^2)$, where n denotes the sentence length and m denotes the average number of POS candidates.

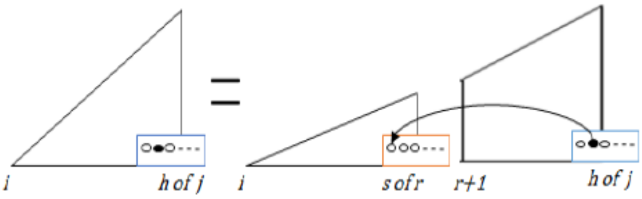
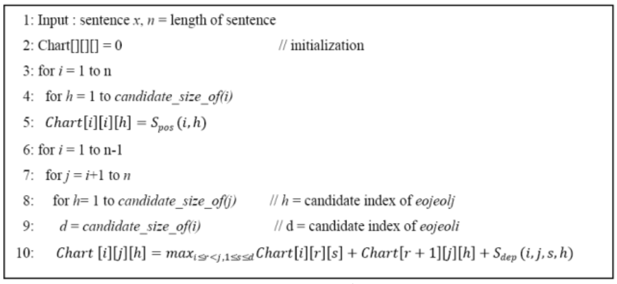


Fig. 3. Dynamic programming structures and derivations of the parsing algorithm.

B. Training Algorithm

To execute the parsing algorithm, we must estimate parameter *w*. We adopt the averaged perceptron [20], [21] that shows a competitive performance in dependency parsing [11]. Here, we modify the averaged perceptron to separately learn the POS and dependency weight, which are shown as pseudo code in Fig. 5. If data sample *n* is a fully tagged datum, all code will be processed. On the contrary, if only the POS is tagged, lines 8 to 12 will be not processed. The POS and dependency part are separated because the dependency result of only POS tagged data causes a noise effect. *T* denotes the number of iterations and vector *v* denotes the accumulated vector. *T* is selected when the best accuracy on the validation set occurs. In line 13, function num(pos) and num(dep) return the number of sentences with tagged POS and dependency.



```
1: Input : sentence $x$, $n$ = length of sentence
2: Chart[][][] = 0                          // initialization
3: for $i$ = 1 to n
4:   for $h$ = 1 to candidate_size_of(i)
5:   $Chart[i][i][h] = S_{pos}(i,h)$
6: for $i$ = 1 to n-1
7:   for $j$ = $i$+1 to $n$
8:     for $h$= 1 to candidate_size_of(j)     // $h$ = candidate index of eojeolj
9:       $d$ = candidate_size_of(i)           // d = candidate index of eojeoli
10:      $Chart\,[i][j][h] = max_{i\le r<j,1\le s\le d} Chart[i][r][s] + Chart[r+1][j][h] + S_{dep}(i,j,s,h)$
```

Fig. 4. Parsing algorithm of the joint model.

C. Feature Functions

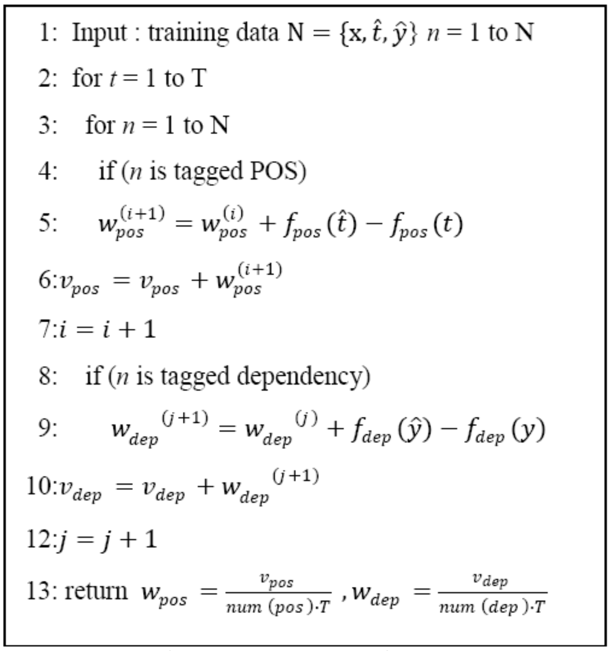
We design feature functions for $f_{pos}(i)$,$f_{dep}(i,j)$ as the following features:

1) POS Features
unigramPOS, bigramPOS, trigramPOS, unigramWord, bigramWord

2) Concatenate Features
lastPOS(i)firstPOS(j), lastWord(i)firstWord(j), last$_{-1}$POS(i)lastPOS(i)firstPOS(j), lastPOS(i)firstPOS(j)first$_{+1}$POS(j)

3) Relation Features
Combinations of POS, Word, L, FPOS, FWord, CPOS, and CWord.

For the above feature functions, POS features are extracted only for $f_{pos}(i)$; other features are extracted for $f_{dep}(i,j)$. The concatenate features are extracted from the form of sequentially concatenated dependent and head text with consideration of the continuity context. The prefix *last* refers to the last morpheme in an *eojeol;* the prefix *first* refers to the first morpheme. The POS denotes a part of speech of the corresponding morpheme, while a word is a whole morpheme. In cases with prefixes *C* and *F*, the feature function is extracted from only content morphemes and function morphemes, respectively. Finally, *L* represents the length of the number of *eojeols* between the dependent and head.



```
1:  Input : training data N = {x, $\hat{t}$, $\hat{y}$} $n$ = 1 to N
2:  for $t$ = 1 to T
3:   for $n$ = 1 to N
4:     if ($n$ is tagged POS)
5:       $w_{pos}^{(i+1)} = w_{pos}^{(i)} + f_{pos}(\hat{t}) - f_{pos}(t)$
6:$v_{pos} = v_{pos} + w_{pos}^{(i+1)}$
7:$i = i + 1$
8:   if ($n$ is tagged dependency)
9:      $w_{dep}{}^{(j+1)} = w_{dep}{}^{(j)} + f_{dep}(\hat{y}) - f_{dep}(y)$
10:$v_{dep} = v_{dep} + w_{dep}{}^{(j+1)}$
12:$j = j + 1$
13: return $w_{pos} = \frac{v_{pos}}{num\,(pos)\cdot T}$ , $w_{dep} = \frac{v_{dep}}{num\,(dep)\cdot T}$
```

Fig. 5. Modified averaged perceptron for the joint model.

V. Experiments

We evaluated the baselines and our models on the Sejong corpus [22] for performance verification. The syntactic part of the Sejong corpus consists of POS tagging and a constituency structure. Therefore, we converted the constituency structures to dependency structures using the head-final rule. We split the corpus using random sampling for training of 80%, validation of 10%, and testing of 10% of the fully tagged corpus. Only the POS tagged corpus was used as the additional training set. To conduct a fair evaluation, only the POS tagged corpus already existing in the fully tagged corpus was removed. A validation set was used for the decision training number of iteration T. In our evaluation, accuracy was measured with an eojeol score (ES) and unlabeled attachment score (UAS) for POS tagging and dependency parsing.

TABLE I: EXPERIMENTAL RESULTS

Model	ES of POS tagging	UAS of Dependency Parsing
Pipelined Baseline	93.83	84.05
Joint Model with Fully tagged corpus	95.69	85.71
Joint Model with Fully + POS corpus	96.17	85.70
Gold POS + Baseline[19]	100	87.67
Han 2005[6]	95.4	-
Park 2011[17]	95.9	-

Table I presents the performance evaluation results. The pipelined baseline description in Section III shows a 93.83% and 84.05% accuracy. This dependency parsing performance represents a 3.6% decrease in accuracy from that of gold POS tagging. However, our proposed model showed improved accuracy of both POS tagging and dependency parsing over the pipelined structure despite not employing the POS tagged corpus. Moreover, in the case of adding all the corpus, dependency parsing accuracy was not improved, whereas POS tagging significantly improved. The evaluation results demonstrate that our joint model solves the error propagation problem of the pipelined structure and improves POS tagging with syntactic information. The methods of Han [6] and Park [7] cannot be directly compared to our model because ours was evaluated on a different corpus. Nevertheless, our evaluation results mean that our model is competitive with recently proposed methods.

VI. CONCLUSION

In this paper, we proposed a joint model of POS tagging and dependency parsing for the Korean language. Experimental results show that our joint model improves performance of both POS tagging and dependency parsing over the pipelined structure for Korean natural language processing. Our model additionally employs a POS tagged corpus not employed in previous works. The evaluation results show that our model is an effective method.

Most state-of-the-art dependency parsing models have adopted higher-order factorization [11]. In this study, we adopted only first-order factorization; therefore, use of higher-order factorization may enable improvement. Moreover, we should apply to a transition-based parsing model our model's property that employs the POS tagged corpus.

ACKNOWLEDGMENT

This work was supported by ATC (Advanced Technology Center) Program – "Development of Conversational Q&A Search Framework Based on Linked Data: Project No. 10048448"

REFERENCES

[1] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in *Proc. the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 118-127.

[2] D. Han, M.-G. Pascual, Y. Miyao, K. Sudoh, and M. Nagata, "Using unlabeled dependency parsing for pre-reordering for Chinese-to-Japanese statistical machine translation," in *Proc. the ACL Second Workshop on Hybrid Approaches to Translation*, 2013, pp. 25-33.

[3] S. Mukherjee and P. Bhattacharyya, "Feature specific sentiment analysis for product reviews," *Computational Linguistics and Intelligent Text Processing*, 2012, pp. 475-487.

[4] J.-F. Gao, J.-Y. Nie, G.-Y. Wu, and G.-H. Cao, "Dependence language Model for information retrieval," in *Proc. the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004, pp. 170-177.

[5] J.-C. Shin and C.-Y. Ock, "A Korean morphological analyzer using a pre-analyzed partial word-phrase dictionary," *Journal of KIISE: Software and Applications*, vol. 39, no. 5, 2012, pp. 415-424.

[6] C.-H. Han and M. Palmer, "A morphological tagger for Korean: Statistical tagging combined with corpus-based morphological rule application," *Machine Translation,* vol. 18, no. 4, 2004, pp. 275-297.

[7] J. Park, J.-W. Cha, and S.-W. Jang, "Korean POS tagging using syllable lattice based OOV words resolution," *Information — An International Interdisciplinary Journal*, vol. 14, no. 8, 2011, pp. 2835-2843.

[8] Y.-H. Lee and J.-H. Lee, "Korean dependency parsing using online learning," *Korea Computer Congress 2010*, 2010, pp. 299-304.

[9] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. the 18th International Conference on Machine Learning*, 2001, pp. 282-289.

[10] R. McDonald, F. Pereira, K. Ribarov, and J/ Hajič, "Non-projective dependency parsing using spanning tree algorithms," in *Proc. the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 523-530.

[11] T. Koo and M. Collins "Efficient third-order dependency parsers," in *Proc. the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 1-11.

[12] J. Nivre, J. Hall, and J. Nilsson, "Maltparser: A data-driven parser-generator for dependency parsing," in *Proc. LREC,* vol. 6, 2006, pp. 2216-2219.

[13] Y. Zhang and J. Nivre, "Transition-based dependency parsing with rich non-local features," in P*roc. the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers,* vol. 2, 2011, pp. 188-193.

[14] R. McDonald and J. Nivre, "Analyzing integrating dependency parsers," *Computational Linguistics*, vol. 37, no. 1, 2011, pp. 197-230.

[15] Z.-H. Li, M. Zhang, W.-X. Che, T. Liu, W.-L. Chen, and H.-Z. Li, "Joint models for Chinese POS tagging and dependency parsing," in *Proc. the Conference on Empirical Methods in Natural Language Processing*, 2011 pp. 1180-1191.

[16] B. Bohnet and J. Nivre, "A transition-based System for Joint part-of-speech tagging and labeled non-projective dependency parsing," in *Proc. the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1455-1465.

[17] Ulsan. [Online]. Available: http://nlplab.ulsan.ac.kr/

[18] D.-M. Kim, Y.-J. Bae, C.-Y. Ock, H.-S. Choi, and C.-H. Kim, "Korean POS and homonym tagging system using HMM," in *Proc. the 20th Annual Conference on Human and Cognitive Language Technology*, 2008, pp. 12-16.

[19] J. M. Eisner, "Three new probabilistic models for dependency parsing: An exploration," in *Proc. the 16th conference on Computational Linguistics*, vol. 1, 1996, pp. 340-345.

[20] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proc. the ACL-02 Conference on Empirical Methods in Natural Language Processing,* vol. 10, 2002, pp. 1-8.

[21] M. Collins and B. Roark, "Incremental parsing with the perceptron algorithm," in *Proc. the 42nd Annual Meeting on Association for Computational Linguistics*, 2004.

[22] Sejong. [Online]. Available: http://www.sejong.or.kr/

Youngmin Park is a Ph.D. student in Sogang University, Korea. He received his B.S degree in computer engineering from Korea Maritime and Ocean University, Korea, in 2009. He received his M.S degree in computer science from Sogang University, Korea, in 2011. His research interests include natural language processing, machine learning and information extraction.

Jungyun Seo is a full professor of computer science at Sogang University. He was educated at Sogang University, where he obtained a B.S. degree in mathematics in 1981. He continued his studies at the Department of Computer Science in the University of Texas, Austin. He receiving a M.S. and a Ph.D. in computer science in 1985 and 1990 respectively. He returned to Korea in 1991 to join the faculty of Korea Advanced Institute of Science and Technology in Taejon where he leaded the Natural Language Processing Laboratory in Computer Science Department. In 1995, he moved to the Sogang University in Seoul and became a full professor in 2001. His research interests include multi-modal dialogues, statistical methods for NLP, machine translation and information retrieval.