J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

C3Tree Model for Learning a Better Korean Sentence☆

Janghwan Kim[1] Woosung Jang[2] R. Young Chul Kim[3*]

ABSTRACT

Recently, there has been a rapid growth in large language models (LLMs), sophisticated AI systems that can understand, generate, and manipulate human language. Most AI experts mention prompt engineering, RAG, or fine-tuning as key factors for achieving success with LLM. Even in the field of requirement engineering, we also handle requirement specifications written in the Korean language. Howe ver, this language has complex grammatical structures and diverse morpheme combinations, which are crucial factors affecting the performance and interpretation of natural language processing (NLP) models. We attempt to analyze the unique and complex gramma tical structures of the Korean language, including complicated, compounded, and simple sentences. As requirement engineers, we sho uld make it challenging for AI machines to understand and process the sentences they learn accurately. To solve this issue, we propose a simple sentence generation mechanism with the C3Tree model. By converting complex sentences into simpler forms, our approach aims to reduce sentence redundancy and also enhance the quality of training sentences. Then, we try to learn the generated simple, compounded, and complicated sentences using AI Machine Learning models to enhance the F1-Score. As a result, we expect to enha nce the accuracy of Korean NLP models and improve data quality, thereby making them more effective for various AI applications.

☞ keyword : Korean Natural Language Requirement Specifications, Textual Analysis, C3Tree, Natural Language Processing

1. Introduction

In this moment, natural language processing (NLP) is a field that enables computers to understand and process human language, playing a key role in various AI applications. However, Korean, one of the natural languages, is a language with complex grammatical structures and diverse morpheme combinations, posing enormous challenges for machines to understand and process it effectively[1]. Thecomplex sentences can create difficulties in morphological and syntactic analysis, negatively impacting the performance of NLP models [2-4]. Therefore, a process called 'simplification' is necessary to convert complex sentences into simpler forms [5].

Simplifying complex sentences helps machines improve text comprehension, enhances the performance of NLP models, and contributes to maintaining data consistency [6].

We are deeply going to consider that Simple sentences are easier to understand, not only for machines but also for humans, making it easier to convey the meaning of information clearly [7]. Additionally, when the structure is simplified, the accuracy of analysis and prediction can be increased, and reducing structural differences within datasets leads to improved quality of training data [8].

We propose a simple sentence generation mechanism to convert complex Korean sentences into simpler sentences, aiming to improve the performance of NLP models. We may expect our approach to enhance the accuracy of Korean NLP models and improve the quality of training data.

Chapter 2 mentions related studies that introduce existing methods for analyzing Korean sentences using machine learning and other techniques. Chapter 3 presents a C3Tree-based Korean sentence refinement mechanism designed to enhancethe quality of Korean sentences. Chapter 4 demonstrates how to construct simple sentences using the

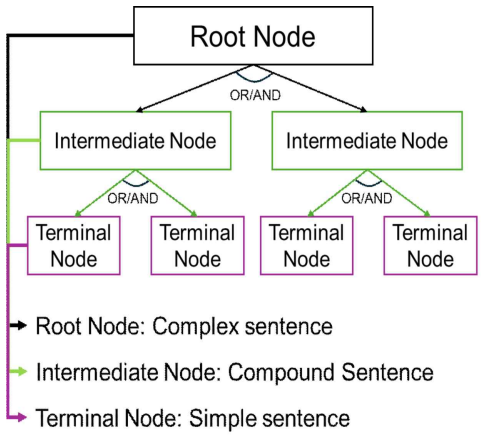
1,3 Dept. of Software and Communications Engineering, Hongik University, Sejong, 30016, Republic of Korea
2 Software Engineering Laboratory, Hongik University, Sejong, 30016, Republic of Korea
* Corresponding author: bob@hongik.ac.kr
[Received 27 July 2025, Reviewed 12 August 2025(R2 10 September 2025, R3 04 November 2025), Accepted 08 November 2025]
☆ This research was conducted with the support of the Korea Creative Content Agency (Project Name: Artificial Intelligence-Based Interactive Multimodal Interactive Storytelling 3D Scene Authoring Technology Development, Project Number: RS-2023-00227917, Contribution Rate: 100%) and the Korea Research Foundation's four, Brain Korea 21 (Project Name: Ultra-Distributed Autonomous Computing Service Technology Research Team, Project Number: 202003520005).
☆ A preliminary version of this paper was presented at ICONI 2024.

C3Tree model and obtain F1-scores. Finally, we mention the conclusion.

2. Related Works

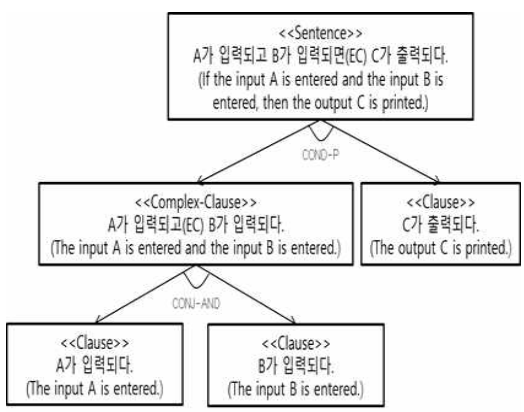
2.1 Natural Language Requirements Analysis with C3Tree Model

The C3Tree stands for Conditional, Conjunction, and Clause Tree, that is, a Korean sentence analysis approach designed to simplify complex sentences [9-11].



(Figure 1) The Definition of the C3Tree Model

Figure 1 illustrates the definition of the C3Tree Model and its operation in a diagrammatic form.



(Figure 2) C3Tree Model Example

Figure 2 shows an example of the requirements analysis process using the C3Tree model.

The C3Tree model analyzes complex sentences by breaking them down into clauses using a tree structure and restores any omitted subjects in the process. It also converts passive sentences into active ones and merges simplified sentences with similar meanings into a single form. Through these steps, the model restructures complex sentences into a more concise and consistent form, thereby helping to resolve ambiguities in the requirements.

2.2 KLUE Benchmark

Korean Language Understanding Evaluation (KLUE) is a benchmark designed to evaluate the performance of Korean Natural Language Processing (NLP) models. It is used to test and assess various tasks related to understanding the Korean language. KLUE encompasses tasks such as sentence classification, sentence similarity, sentiment analysis, question answering, and natural language inference. Evaluating these tasks helps measure how well Korean NLP models understand and process language [12]. This benchmark is a crucial tool for objectively comparing how models analyze complex Korean sentences, and it also highlights the importance of high-quality training data in the Korean language.

(Table 1) KLUE Benchmark Exam Categories

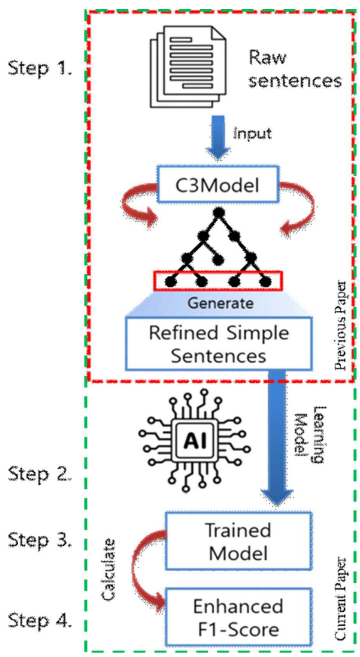
Name	Description
Topic Classification (TC)	This provides a classifier for predicting the topic of text snippets.
Semantic Textual Similarity (STS)	This measures the degree of semantic equivalence between two sentences.
Natural Language Inference (NLI)	This reads pairs of whole sentences and hypothesis sentences and predicts whether the relationship is contradictory or neutral.
Named Entity Recognition (NER)	This detects the boundaries of named entities in unstructured text and classifies them by type.
Relation Extraction (RE)	This identifies semantic relations between entity pairs in a text.
Dependency Parsing (DP)	This finds relational information among words.

Machine Reading Comprehension (MRC)	This assesses the ability to comprehend questions and locate answers.
Dialogue State Tracking (DST)	This predicts the dialogue states from a given dialogue context.

Table 1 shows the eight categories covered by KLUE. For semantic analysis, simplified sentences provide clear emotional signals, which can improve model performance. Simplified sentences express information more clearly, which can lead to more accurate and consistent answers.

3. Korean Sentence Simplification Mechanism

We use complex sentences as input and prompt them into the C3Tree model to generate compound sentences. And then, it will break down to several simplified sentences.



(Figure 3) Enhancing Model Learning Process

Figure 3 illustrates our proposed method for enhancing the F1 score of a trained model. A common challenge in simplifying complex sentences is the inconsistent subjects that often result. We address this issue by incorporating the subject restoration method proposed by Jang et al. [8]. This approach yields simplified sentences with a more consistent structure, thereby improving their interpretability and making them easier for the model to classify.

We're applying this technique to extract simple sentences and compound sentences from existing original complex sentences. In our case study, we will explain the entire proposed process. We'll be comparing the F1 scores within the Topic Classification Category of the KLUE Benchmark exam.

4. Learning Results with Our Approach

To construct our dataset, we began with 2000 complex sentences and generated their simplified counterparts using the C3Tree model. This resulted in a total of 4000 sentences, comprising the original 2000 complex sentences and the 2000 newly created simple sentences, which were then used for training.

With the extracted sentences from the C3Tree model, we train them using various AI Machine Learning Algorithms. We aim to identify a more effective sentence style that is better understood by a particular learning algorithm. Following this conversion, we measure the F1 score according to the proposed process.

Step 1. Simplifying complex sentences by using the C3Tree Model:

First, a complex sentence is input into the C3Tree model analyzer. Then, the analyzer converts the complex sentence into a short sentence.We will show three different examples.

<Case 1>:

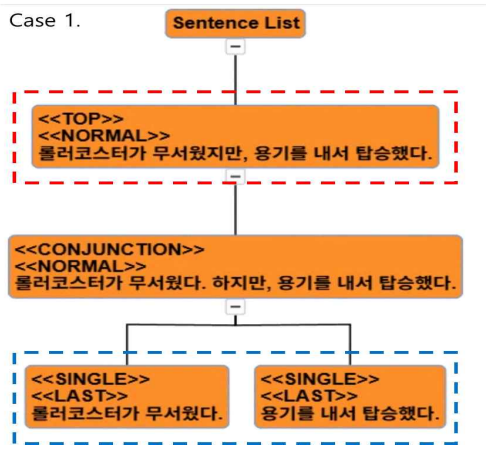
This case is when a sentence contains two contrasting or inverse relationships. This case is when the contents of two contrasting or inverse relationships are included in one sentence. When we convert an input sentence to a compound sentence, coordinating conjunctions such as 'but' or 'however' are used to clarify the meaning of the contrast. We eliminate the conjunctions from the input sentence. And then, it breaks down into simple sentences, which align with the expected

result in Table 2.

(Table 2) Input & Expected Output Sample of Case 1

Case 1	Korean Sentence
	English Sentence
Input	롤러코스터가 무서웠지만, 용기내서 탑승했다.
	Although the roller coaster was scary, I rode it.
Expected Result	1. 롤러코스터가 무서웠다. 2. 용기내서 탑승했다.
	1. The roller coaster was scary. 2. I rode it.

Figure 4 shows the analysis result through the C3Tree model analyzer. The red box contains the original text, which is a complex sentence, and is converted into a simpler sentence, like the blue box,through the rule-based algorithm of the analyzer.



(Figure 4) C3Tree Transformation Example for Case 1

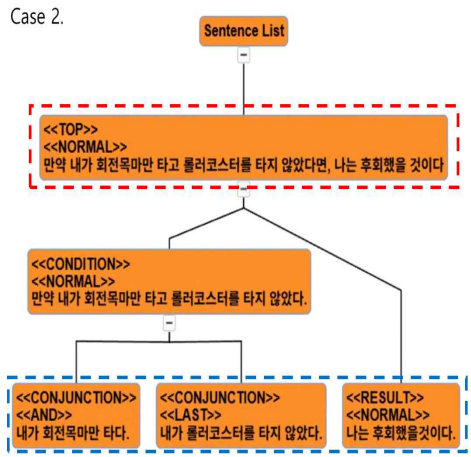
<Case 2>:

This case is an example of a sentence that contains two contrasts or inverse relationships. This case is a form in which a subordinate clause expressing an assumption or condition is combined with a main clause. When converting to a compound sentence, an inverse conjunction such as '~jiman' or 'but' may be implied to connect two independent facts while preserving the meaning of the assumption.

(Table 3) Input & Expected Output Sample of Case 2

Case 2	Korean Sentence
	English Sentence
Input	만약 내가 회전목마만 타고 롤러코스터를 타지 않았다면, 나는 후회했을 것이다.
	If I had only ridden the carousel and not the roller coaster, I would have regretted it.
Expected Result	1. 내가 회전목마만 타다. 2. 내가 롤러코스터를 타지 않았다. 3. 나는 후회했을 것이다.
	1. I only rode the carousel. 2. I didn't ride the roller coaster. 3. I would have regretted it.

Figure 5 shows the analysis result of case 2 through the C3Tree model analyzer. The red box represents the original text, a complex sentence, which is converted into a simple sentence, like the blue box, through the analyzer.



(Figure 5) C3Tree Transformation Example for Case 2

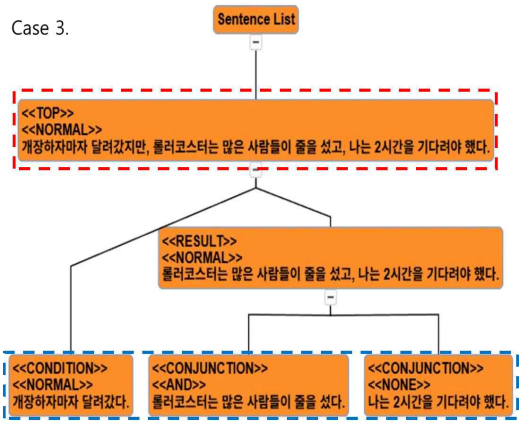
<Case 3>:

This case is an example of a sentence that contains two contrasting or inverse relationships. This case is a complex sentence with a subordinate clause expressing the temporal order or continuity of actions or events ('-go', '-ha-ja-ma-ja'). When converting to a compound sentence, a coordinating conjunction expressing a sequential relationship, such as 'and', is mainly used.

(Table 4) Input & Expected Output Sample of Case 3

Case 3	Korean Sentence English Sentence
Input(Complex Sentence)	개장하자마자 달려갔지만, 롤러코스터는 많은 사람들이 줄을 섰고, 나는 2시간을 기다려야 했다. Though I ran there as soon as it opened, there was a long line for the roller coaster, and I had to wait for two hours.
Expected Result(Simple Sentence)	1. 개장하자마 달려갔다. 2. 롤러코스터는 많은 사람들이 줄을 섰다. 3. 나는 2시간을 기다려야 했다. 1. I ran there as soon as it opened. 2. There were a lot of people lining up for the roller coaster. 3. I had to wait for 2 hours.

Figure 6 shows the analysis result of case 3, using the C3Tree model analyzer. The red box contains the original text, which is a complex sentence, and it is converted into a simple sentence.



(Figure 6) C3Tree Transformation Example for Case 3

Step 2. Data preparation & preprocessing

Then, we extracted the 'label' from the selab_data source and divided it into a 'complex sentence' column and a 'single sentence' column, so that the model could classify the meaning of the two types of sentences. Then, we created and assigned each label and two different data frames, complexSentenceData and SingleSentenceData.

(Table 5) Stop word table

Korean	Roman Character	Meaning
합니다	hamnida	formal ending for 'do' or 'be'
하는	haneun	present participle form of 'do'
할	hal	future participle form of 'do'
하고	hago	conjunctive form of 'do', and'
한다	handa	plain form of 'do'
그리고	geurigo	and, then
입니다	imnida	formal ending for 'is/am/are'
그	geu	that, the pronoun for 'he/she/it'
등	deung	etc., and so on, others
이런	ireon	this kind of, such
것	geot	thing, fact, what
및	mit	and, as well as
제	je	my, I - humble form
더	deo	More

To ensure data completeness, we removed all rows with missing values in the 'complex sentence' or 'single sentence' columns from each data frame. We performed a stopwordremoval process to process text data efficiently. We defined a list of Korean stop words such as '합니다' (hamnida), '하는' (haneun, '할' (hal), '하고' (hago), '한다' (han-da), '그리고' (geu-ri-go), '있다' (ib-ni-da), '그' (gue), '등' (deung), '이런' (ee-reon), '것' (geot), '과' (mit), '제' (je), '더' (deo), and generated preprocessed text ('sentences_prep').

We split the preprocessed data into training and test sets. For each dataset (single sentence and complex sentence), we divided the data into a 75% training set and a 25% test set. We used the stratify option to maintain the balance of the label distribution and fixed the 'random_state' to 42 to ensure reproducibility.

Finally, to convert the text data into numerical features that the machine learning model can understand, weapplied TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. We created featurevectors using TfidfVectorizer with the N-gram range set to (1, 3), the minimum document frequency (min_df) set to 3, and the maximum document frequency (max_df) set to 0.95.

Step 3. Training with fundamental machine learning Algorithms.

We selected some widely used machine learning models for text classification and evaluated their performance. The chosen models are Random Forest Classifier, Support Vector

Machine, and Logistic Regression. XGBoost and LightGBM were excluded from the evaluation.

The performance of each model was evaluated using the 5-fold cross-validation method. This is to mitigate the bias introduced by a specific data split and to measure the model's generalization performance more accurately. In the cross-validation process, each fold evaluates the model using a validation set separated from the training set.

As model performance indicators, we used Accuracy, Precision, Recall, and F1-Score. In particular, considering that it is a multi-class classification problem, we applied the 'average=" macro "' option when calculating Precision, Recall, and F1-Score to reflect the scores for each class equally and then calculated the average.

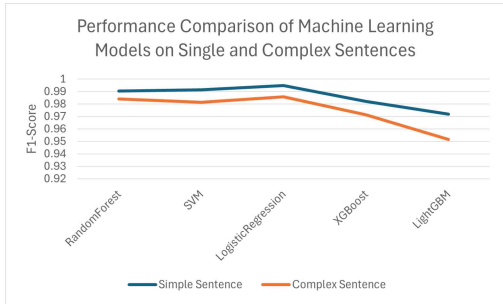
Step 4. Evaluate and compare AI models

We collected and analyzed the cross-validation results of models for each dataset (single sentence and complex sentence) using the evaluate_models function. We compared the performance of each model by displaying the average accuracy, precision, recall, F1-score, and other metrics up to the fourth decimal place. The performance indices for each model for the single sentence dataset and the complex sentence dataset are as shown in Table 6.

A direct comparison reveals a clear trend: overall performance metrics (such as Accuracy, Precision, Recall, and F1-Score) are higher for models trained on the simple sentence dataset.

(Table 6) Performance Result

		Random Forest	SVM	Logistic Regression	XGBoost	Light GBM
Complex Sentence	Accuracy	0.9893	0.9873	0.9893	0.9787	0.9620
	Precision	0.9917	0.9933	0.9958	0.9847	0.9776
	Recall	0.9904	0.9915	0.9954	0.9830	0.9741
	F1Score	0.9903	0.9914	0.9948	0.9821	0.9719
CP	⋮	⋮	⋮	⋮	⋮	⋮
Simple Sentence	Accuracy	0.9959	0.9946	0.9953	0.9865	0.9750
	Precision	0.9860	0.9837	0.9870	0.9728	0.9578
	Recall	0.9857	0.9834	0.9878	0.9751	0.9570
	F1Score	0.9841	0.9813	0.9857	0.9712	0.9515



(Figure 7) Performance Comparison Chart of Machine Learning Models on Single and Complex Sentences.

As Figure 7 shows,across all machine learning algorithms tested—Random Forest, SVM, Logistic Regression, XGBoost, and LightGBM—the F1-Score for simple sentences was higher than that for complex sentences. This suggests that simplifying sentence structures into individual, atomic units before training can enhance the classification performance of these models.

First, all models showed high accuracy above 0.96, indicating that the dataset quality and feature design were stable. In particular, most models achieved an accuracy higher than 0.98 on the single-sentence dataset and still maintained over 0.96 on the complex-sentence dataset. This means that each model has a consistent level of generalization performance regardless of sentence structure or length. Therefore, the models were not overfitted to a specific sentence type but demonstrated an overall understanding of sentence-level semantics.

Second, as sentence structures became more complex, all models showed a slight decrease in performance. When moving from single to complex sentences, the average accuracy dropped by about 0.5–2%. LightGBM showed the largest decrease, about 1.3%, suggesting that tree-based models are relatively weak in capturing nonlinear dependencies within complex sentences. This performance drop can be explained by the fact that main and subordinate clauses, conjunctions, and other grammatical connections make the boundary between meaning units more ambiguous.

Third, among the algorithms, Logistic Regression achieved the highest performance on single sentences. This result shows that linear models work more effectively when feature

correlations are simple. In contrast, Random Forest and SVM maintained stable performance even with complex sentences. Random Forest is strong in handling nonlinear patterns through the ensemble effect of multiple decision trees, while SVM shows robust classification through hyperplane-based boundary optimization. However, XGBoost and LightGBM showed a noticeable drop in performance on complex sentences. This suggests that boosting-based models rely more on feature separability than on contextual relationships.

Fourth, both Precision and Recall remained above 0.95 for most models, indicating that the dataset was well-balanced. However, a slight decrease in Precision was observed for complex sentences. This may be because models behaved more conservatively in predicting positive classes when processing polysemous or subordinate structures. In other words, models likely adjusted their decision boundaries more strictly to reduce false positives. This trend reflects a learning tendency aimed at minimizing incorrect positive predictions.

The results highlight the potential challenges models face when processing the increased complexity and interdependencies present in complex sentences. Single sentences, being more direct and less ambiguous, likely provide clearer features for the TF-IDF vectorizer and, subsequently, for the classifiers to learn from. This improved clarity in features appears to contribute to better model generalization and classification accuracy.

5. Conclusion

Recently, there has been a rapid growth in large language models (LLMs), sophisticated AI systems that can understand, generate, and manipulate human language. Most AI experts mention prompt engineering, RAG, or fine-tuning as key factors for achieving success with LLM. Even in the field of requirement engineering, we also handle requirement specifications written in the Korean language. However, this language has complex grammatical structures and diverse morpheme combinations, which are crucial factors affecting the performance and interpretation of natural language processing (NLP) models. We attempt to analyze the unique and complex grammatical structures of the Korean language, including complicated, compounded, and simple sentences.

We propose a rule-based mechanism to convert complex Korean sentences into simpler ones, improving the accuracy and consistency of training data for Korean natural language processing models. By using a C3Tree-based mechanism, we enhance data quality and model performance, reducing difficulties in morphological and syntactic analysis.

As demonstrated by the experimental results, we recognized that models trained on single sentences consistently achieved superior performance across all evaluation metrics (Accuracy, Precision, Recall, and F1-Score) compared to those trained on complex sentence data. We expect to validate the effectiveness of the proposed mechanism by applying it to other domains and datasets, and hope that integrating artificial intelligence with algorithms will lead to further improvements in Korean natural language processing.

References

[1] H. Hwang, H. Kim, “Korean Syntactic Complexity Analyzer (KOSCA): An NLP application for the analysis of syntactic complexity in second language production,” Language Testing, vol.41, no.3, pp.506-529, 2024. https://doi.org/10.1177/02655322231222596

[2] Bates, L, “Korean Sentence Complexity Reduction for Machine Translation,” Doctoral dissertation, Seoul National University Graduate School, 2017.

[3] X. Zhang, M. Lapata, “Sentence simplification with deep reinforcement learning,” arXiv preprint arXiv:1703.10931, 2017. https://doi.org/10.48550/arXiv.1703.10931

[4] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: state of the art, current trends and challenges,” Multimedia Tools and Applications, vol.82, pp.3713-3744, 2023. https://doi.org/10.1007/s11042-022-13428-4

[5] W. Alkaldi, D. Inkpen, “Text simplification to specific readability levels,” Mathematics, vol.11, no.9, 2063, 2023. https://doi.org/10.3390/math11092063

[6] A. Siddharthan, “A survey of research on text simplification,” ITL-International Journal of Applied Linguistics, vol.165, no.2, pp.259-298, 2014. https://doi.org/10.1075/itl.165.2.06sid

[7] G. Song, S. Lee, “A study on the social relationship between humans and machines due to technological evolution,” in Proc. of the 2017 Fall Conference of the Korean Society for Technology Innovation, vol.2017, no.11, pp.425-438, 2017.

[8] Z. Kozareva, Q. Li, K. Zhai, and W. Guo, “Recognizing Salient Entities in Shopping Queries,” in Proc. of the 54th Annual Meeting of the Association for Computational Linguistics, vol.2, pp.107-111, Berlin, Germany, 2016.

[9] W. S. Jang, R. Y. C. Kim, “Automatic cause‐effect graph tool with informal Korean requirement specifications,” Applied Sciences, vol.12, no.18, 9310, 2022. https://doi.org/10.3390/app12189310

[10] W. S Jang, R. Y. C. Kim, “Automatic Generation Mechanism of Cause-Effect Graph with Informal Requirement Specification Based on the Korean Language,” Applied Sciences, vol.11, no.24, 11775, 2021. https://doi.org/10.3390/app112411775

[11] W. Jang, R. Y. C. Kim, “Automatic Test Case Generation Mechanism with Natural Language-based Korean Requirement Specifications,” IEEE Access, vol.13, pp.177305-177317, 2025. https://doi.org/10.1109/ACCESS.2025.3620431

[12] S. Park, et al., “KLUE: Korean Language Understanding Evaluation,” arXiv preprint arXiv:2105.09680, 2021. https://doi.org/10.48550/arXiv.2105.09680

◐ 저 자 소 개 ◑

Janghwan Kim

2019: Received the B.S. degree in Computer Science from Idaho State University, Pocatello, ID, USA.
2022: Received the M.S. degree in Dept. of Software and Communications Engineering from Hongik University, Seoul, South Korea.
2024: Completed the doctoral coursework (Ph.D. candidate) in Dept. ofSoftware and Communications Engineering, Hongik University, Seoul, South Korea.
2024 - Present: Adjunct Professor, Department of Software Convergence, Hongik University, Seoul, South Korea.
Research Interests (ongoing): AI Software Validation, Reinforcement Learning-based Software Validation, Software Visualization, and Requirements Engineering.
e-mail: lentoconstante@hongik.ac.kr

Woosung Jang

2022: Received the Ph.D. degree in Software Engineering from Hongik University, Seoul, South Korea.
2019 - Present: Adjunct Professor, Dept. of Software and Communications Engineering, Hongik University, Seoul, South Korea.
Research Interests (ongoing): Requirements analysis, Korean-Language Requirements modeling, Model-based testing, Metamodeling, and embedded programming.
e-mail: uriel200@hongik.ac.kr

R. Young Chul Kim

1985: Received the B.S. degree in Computer Science from Hongik University, Republic of Korea.
2000: Received the Ph.D. degree in Software Engineering from the Department of Computer Science, Illinois Institute of Technology (IIT), Chicago, USA.
2000‐Present: Professor at Dept. of Software and Communications Engineering Hongik University, Seoul, South Korea
Research Interests (ongoing): Test maturity models, Model-based testing, Metamodeling, Software process models, Software visualization, and validation techniques for Reinforcement Learning software.
e-mail: bob@hongik.ac.kr