Received: 15 June 2020 | Revised: 29 October 2021 | Accepted: 15 March 2022

DOI: 10.4218/etrij.2020-0245

ORIGINAL ARTICLE

ETRI Journal WILEY

Sentence model based subword embeddings for a dialog system

Euisok Chung | **Hyun Woo Kim** | **Hwa Jeon Song**

Integrated Intelligence Research Section, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea

Correspondence
Euisok Chung, Integrated Intelligence Research Section, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea.
Email: eschung@etri.re.kr

Funding information
Electronics and Telecommunications Research Institute (ETRI), Grant/Award Number: 22ZS1100

Abstract

This study focuses on improving a word embedding model to enhance the performance of downstream tasks, such as those of dialog systems. To improve traditional word embedding models, such as skip-gram, it is critical to refine the word features and expand the context model. In this paper, we approach the word model from the perspective of subword embedding and attempt to extend the context model by integrating various sentence models. Our proposed sentence model is a subword-based skip-thought model that integrates self-attention and relative position encoding techniques. We also propose a clustering-based dialog model for downstream task verification and evaluate its relationship with the sentence-model-based subword embedding technique. The proposed subword embedding method produces better results than previous methods in evaluating word and sentence similarity. In addition, the downstream task verification, a clustering-based dialog system, demonstrates an improvement of up to 4.86% over the results of *FastText* in previous research.

KEYWORDS
dialog, embedding, sentence model, subword

1 | INTRODUCTION

Recently, large-scale transfer learning has produced promising results in deep-learning-based language processing [1–3], indicating that large learning parameters and long learning times are essential to system performance. However, further research is required to apply large-scale transfer learning to dialog systems. This is because modern dialog systems depend on multiple domains, knowledge bases, and modalities and thus require more detailed approaches. Word embedding techniques that can be easily used in various network designs are a potential solution. In particular, [4] and [5] used word embedding techniques in sentence generation to obtain impressive results. Specifically, [4] integrated word embeddings into the input/output space to capture and utilize the output structure of the language. In addition, [5] used independently learned word embeddings to train a sequence model. The word embedding technique played the role of a teacher network in the knowledge distillation training method, which is a type of neural network learning method. In [5], the performance of word embeddings had a great influence on the performance of the sequence model.

In this paper, we attempt to replace the sequence model with a dialog model. We create a target based

on word embeddings in the dialog model and then train and evaluate the dialog model based on this. To this end, we first focus on improving the performance of word embedding technology. We propose a new subword embedding technique that considers sentence information and utilizes a clustering-based dialog system methodology for verification. A typical word embedding approach, such as the skip-gram model [6], employs high-capacity text as learning data. However, a problem with the skip-gram technique is that it learns word embeddings only for a given lexical list. This problem is solved by a method called subword embedding, which expresses a given word as a set of specific sub-features and performs embedding to learn the sub-features [7]. Another problem is to determine how to augment word vector models for the task of sentence representation. The reason for needing to augment the word vector models is that most language processing systems require sentence context processing. In particular, for a dialog system, a word embedding technique capable of matching the input and output sequences of sentences is essential, and the clustering-based dialog system proposed in this study can be a means of verifying the word embedding technique.

The proposed word embedding technique is based on subword information and integrates the sentence model by using the skip-thought model [8]. In this study, various experimental results are presented in the sentence model, and self-attention [9] and relative position encoding [10] are considered. Korean is used as the target language; therefore, the position encoding technique for Korean syntactic words (*Eojeol*) is also verified.

The clustering-based dialog system can be used as a verification platform for various word embedding technologies. The dialog set response list is converted into sentence representations through word embedding, and response clusters are generated by *K*-means clustering [11]. By encoding a unified sequence of the given context and question, the decoder responds with the correct sentence and simultaneously predicts the response class. This is useful for validating word and sentence embedding techniques suitable for a dialog system. In particular, in the case of dialog domains, the scope of responses is limited. Therefore, a clustering approach based on sentence embeddings for the corresponding responses is available. It is thus considered that the dialog response prediction performance is closely related to the sentence embedding performance. We constructed 7236 conversation sets[1] from a Korean dialog-based clothing recommendation domain to evaluate the dialog system.

[1]https://fashion-how.org/ETRI/board.html.

The contributions of this study are as follows:

- This study demonstrates that the proposed subword embedding technique using sentence information from various evaluation sets leads to improved results compared with existing techniques. The evaluation set consists of the lexical level, sentence level, and dialog context level.
- The clustering-based dialog system indicates the importance and utilization of word embedding technology.

In Section 2, we discuss related work and describe subword embedding techniques and dialog models. Finally, we provide various experimental results and conclusions.

2 | RELATED WORK

2.1 | Word embedding approaches

Recent research related to word embedding involves the application of text mining and improvements to word embedding. Specifically, research on text mining to extract latent knowledge from large volumes of data in scientific journals has recently been conducted [12]. In addition, studies on improving word embedding have proposed changes to the context model through dependency structuring [13], multi-meaning embedding [14,15], a meta-embedding technique for integrating various word embeddings [16], and out-of-vocabulary problem-solving [7,17,18]. This study starts by reviewing word embedding technology that uses subwords to solve out-of-vocabulary problems.

In a previous study [19], FastText [7] exhibited more stable subword embedding performance than byte-pair embeddings [20]. In the present study, we integrate a sentence model into FastText, and the sentence model integration follows the skip-thought [8] structure. However, the sentence model is different from the skip-thought model because instead of performing sentence embedding using long short-term memory [21], the sentence model is composed of only subword parameters. Additionally, similar to the skip-gram context model, similar sentences within a specific distance are selected as the context sentence. Word embedding using our sentence model is similar to the Siamese CBOW model [22]; namely, it embeds the surrounding sentences and target sentences into the constituent word embedding average values and reflects the cosine similarity of the sentence embedding pairs to word embedding learning. However, we use the

skip-gram model instead of the CBOW model for the sentence model and use subwords as the embedding unit. In terms of multi-prototype word embeddings, we do not model directly as in previous studies [14,15] but take an approach that considers the multi-meaning at the point of application of word embedding. We achieve this approach through self-attention and relative position encoding.

In this study, Korean is the target language. In subword embedding for English, the subword features can be composed of alphabetic characters. Similarly, the subword embedding for Korean can use syllables (characters), phonemes, and graphemes. According to a previous study [23], there was no significant difference between syllable and phoneme subword results. Furthermore, in [17], syllable and grapheme subword features exhibited similar results; however, by using two features in combination, an increase in the word embedding performance was reported. In this study, we use syllables as subword features because the type of subword feature is not the focus of this paper. We use the evaluation sets provided by [17] for the evaluation of word similarity and word analogy. In addition, we construct a sentence similarity evaluation set and examine the relationship between word embedding and the dialog system using the dialog evaluation set.

2.2 | Dialog model approaches

A dialog system can be largely classified into a task-based approach and a non-task-based approach [24]. However, even for the task-based approach, most recent research has followed an end-to-end approach [25,26]. In the field of response generation, retrieval-based [27–29], generation-based [30], and hybrid-based [31–33] approaches have been studied. Recently, approaches to reduce the search space of a conversation interaction by clustering the components of the conversation set have been attempted [34–36]. In particular, [35] performed clustering through utterance embedding and proposed a method of tracking and predicting an utterance class. The dialog model in the present study is similar to the approach proposed in [35]. However, the difference is that we cluster only the response results and employ a predictive approach when decoding. In addition, a knowledge network related to the dialog context is added to the encoder, while for multi-task learning, a response generation function is added to the decoder. This study focuses mainly on the relationship between word embedding and dialog modeling.

3 | SUBWORD MODEL

The main topic of this paper is how to project a sentence context into subword vectors. The skip-gram model [37] is a prototype of the proposed model and describes a probability model in which the target word predicts nearby words. In this study, we perform word embedding by extending the skip-gram model to a subword-based approach, referred to as the subword skip-gram model (SSGM). Then, sentence modeling is performed using a subword skip-thought model (SSTM).

3.1 | Integration of the sentence model

The structure of the two models is presented in Figure 1, which illustrates the connection between the two models with the shared subword parameters Φ_t. Equation (1) describes a model that integrates SSGM and SSTM. The goal of this study is to determine the subword vectors Φ_t that maximize the log-likelihood $\mathcal{L}$. Here, $\mathcal{T}_w$ denotes the size of the subword embedding learning data W, while $\mathcal{T}_s$ denotes the size of the skip-thought learning data S. $\mathcal{C}_t$ is a set of context words w_c for the target word w_t, while $\mathcal{N}_t$ is a set of context sentences s_n for the target sentence s_t.

$$\mathcal{L} = \sum_{t=1}^{T_w} \sum_{c \in C_t} \log\, p(w_c|w_t) + \sum_{t=1}^{T_s} \sum_{n \in N_t} \log\, p(s_n|s_t). \quad (1)$$

SSGM. Equation (2) is an exponential language model using the score function q of w_t and w_c. The existing subword model [7] uses only the constituent features $\mathcal{F}_t$ of the target word w_t in (3), and the context word w_c uses the word vector v_c. In (3), f is one of the many subword features of w_t. $\mathcal{F}_t$ is a set of the subword features of w_t, and z_f is the subword vector for f. Thus, $f \in \mathcal{F}_t$ and $z_f \in \Phi_t$.

$$p(w_c|w_t) = \frac{e^{q(w_t,w_c)}}{\sum_{i=1}^{\mathcal{T}_w} e^{q(w_t,w_i)}}, \quad (2)$$

$$q(w_t,w_c) = \sum_{f \in \mathcal{F}_t} z_f^{\top} \cdot v_c. \quad (3)$$

We modify v_c in (3) as the sum of subword vectors z_g to reduce the computational burden because the number of vocabularies is generally very large. Equation (4) describes these modifications. It makes a relation, $f \in \mathcal{F}_c$

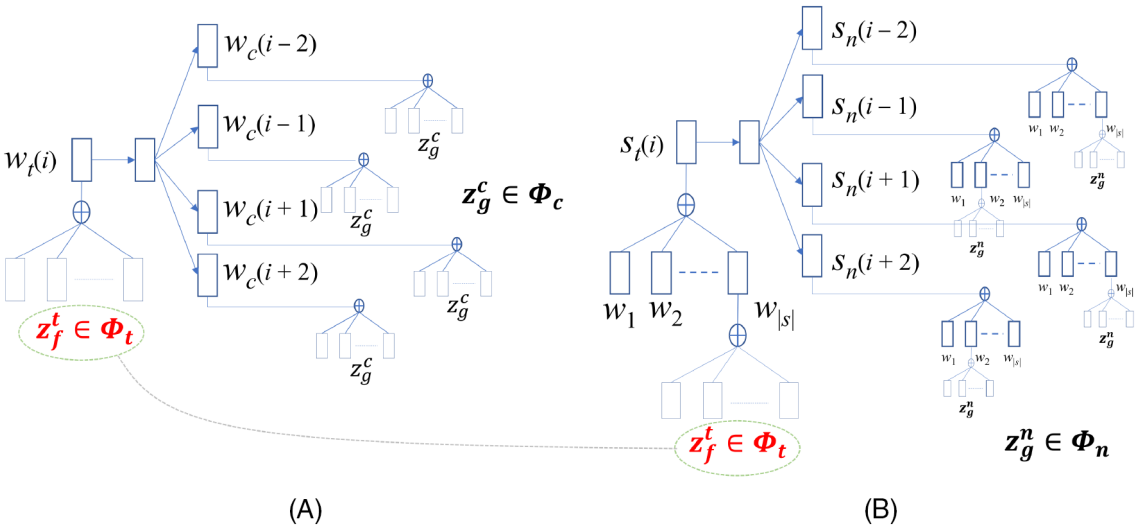

FIGURE 1 (A) Subword skip-gram model (SSGM) and (B) subword skip-thought model (SSTM). Here, Φ_t is a set of subword vectors for the SSGM target word and SSTM target sentence, while Φ_c is a set of subword vectors for the SSGM context word. Φ_n is for the SSTM target sentence. In addition, z_f^t is the subword vector. The subscript f is the subword feature of the target word. The subword features are the same in all three tables, Φ_t, Φ_c, and Φ_n, with the vector values of the subword features being learned simultaneously. In the verification of the learned subword embeddings, we conduct an experiment using only Φ_t (*CV*) and an experiment integrating Φ_t, Φ_c, and Φ_n (*IV*) (see Section 5.1)

and $z_g \in \Phi_c$. The score function q of (4) is a dot product of two vectors, each using the sum of the subword vectors.

$$q(w_t, w_c) = \left(\sum_{f \in \mathcal{F}_t} z_f^t\right)^\top \cdot \left(\sum_{g \in \mathcal{F}_c} z_g^c\right). \tag{4}$$

SSTM. Equation (5) describes a skip-thought model in which the lexical object of a skip-gram model is converted into a sentence object. This differs from the previous approach [8] because we select the bilinear model of sentence vectors through subword embeddings rather than a recurrent neural network (RNN). Thus, unlike the case of an RNN, all learning results are stored in the target Φ_t and context subword vectors Φ_n.

$$p(s_n|s_t) = \frac{e^{q(s_t, s_n)}}{\sum_{i=1}^{T_s} e^{q(s_t, s_i)}}. \tag{5}$$

Equation (6) describes the score function of a target sentence s_t and context sentence s_n. Each sentence vector is calculated as an average of the constituent word vectors. Because the word vector is calculated as the sum of the constituent subword vectors, the learning result is reflected in Φ_t and Φ_n.

$$q(s_t, s_n) = \left(\frac{\sum_{w_t \in s_t} \sum_{f \in \mathcal{F}_t} z_f^t}{|s_t|}\right)^\top \cdot \left(\frac{\sum_{w_n \in s_n} \sum_{g \in \mathcal{F}_n} z_g^n}{|s_n|}\right). \tag{6}$$

For learning, the computational complexity of the denominators in (2) and (5) is excessive. The general solution is to convert the problem into a binary classification problem, such as Monte Carlo noise-contrastive estimation [38]. Negative samples for w_t and s_t are generated using negative models q and q', respectively, and the equation for modeling the binary classification is described in (7). Learning is performed by maximizing this formula.

$$\begin{aligned}\mathcal{L}_{\mathrm{NCE}_k}^{\mathrm{MC}} = &\sum_{(w_c, w_t) \in W} \left(\log p(D=1|w_c, w_t) + \sum_{i=1, \overline{w_t} \sim q}^{k} \log p(D=0|w_c, \overline{w_t})\right) \\ &+ \sum_{(s_c, s_t) \in S} \left(\log p(D=1|s_c, s_t) + \sum_{i=1, \overline{s_t} \sim q'}^{k} \log p(D=0|s_c, \overline{s_t})\right).\end{aligned} \tag{7}$$

3.2 | Extension of the sentence model

In the SSTM, the sentence model uses the mean of the word vector values and combines a variety of approaches. We extend the model according to self-attention [9], relative position encoding [10], and Korean sentence structure.

Self-attention. We utilize the query–key–value ($\mathcal{QKV}$) self-attention described in [9]. First, the sentence is converted into a matrix $\mathcal{S}$ of word vectors, and the matrix is set to a constant value for each $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$. Here, the attention operation, $\text{softmax}(\mathcal{QK}^\top/\sqrt{d_k})\mathcal{V}$, is performed, and the matrix $\mathcal{V}$ is updated using the result. In this process, we do not use additional learning parameters, such as linear layers for $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$. Finally, the average of $\mathcal{S}$ and the self-attention result $\mathcal{V}$ is used.

Relative position encoding. The authors of [10] proposed two relative position encoding parameters, rpe^v and rpe^k, which are integrated with the self-attention operations. In the softmax operation, rpe^v is added to $\mathcal{V}$ according to the relative distance of words, and rpe^k is added to $\mathcal{K}$. Both values consist of n vectors according to the relative distance n, a hyperparameter.

Let an element α_{ij} of $\text{softmax}(\mathcal{QK}^\top/\sqrt{d_k})$ be the attention weight of the jth word vector with respect to the ith word vector of the sentence matrix $\mathcal{V}$. In this case, $\text{rpe}^v_{d(i,j)}$ is added to $\mathcal{V}_j$ according to the relative distance $d(i,j)$ between $\mathcal{V}_i$ and $\mathcal{V}_j$, and the attention operation is performed. Here, $\text{rpe}^k_{d(i,j)}$ is a value added to $\mathcal{K}_j$ according to the relative distance $d(i,j)$ during the $\mathcal{QK}^\top$ operation.

Sentence model for Korean. Because Korean is an agglutinative language, grammatical morphemes are integrated into lexical morphemes to form syntactic words called *Eojeol*.[2] The Korean sentence model adds an *Eojeol* layer, which is an intermediate step between a word and sentence that assigns weights according to positions of the words constituting the *Eojeol*. An *Eojeol* is described as e in (8), where the weight of the word is expressed by η. Because the number of words in each *Eojeol* is different, the first lexical morpheme receives the highest value, while the remaining morphemes receive the same value.

$$q(s_t,s_n) = \left(\frac{\sum_{e_t\in s_t}\sum_{w_t\in e_t}\left[\eta^{s_t}_{w_t}\cdot\sum_{f\in F_t} z^t_f\right]}{|s_t|^e}\right)^\top \cdot \left(\frac{\sum_{e_n\in s_n}\sum_{w_n\in e_n}\left[\eta^{s_n}_{w_n}\cdot\sum_{g\in F_n} z^n_g\right]}{|s_n|^e}\right). \quad (8)$$

[2]Two *Eojeols* are provided as an example. "다로 가" can be read as "*dari-ro ga*" and translated to English as "go to the bridge." Here "다로(*dari-ro*)" means "to the bridge" and "가(*ga*)" means "go." Furthermore, "다(*dari*)" means "bridge," and "-로(*-ro*)" is a postpositional particle.

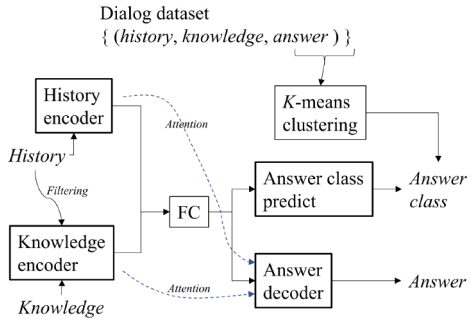

FIGURE 2 Clustering-based dialog model. The history, knowledge, and response are represented as subword embeddings. FC, fully connected

4 | CLUSTERING-BASED DIALOG MODEL

The purpose of the dialog model is to evaluate the subword embedding approach. From another point of view, we examine the relationship between utterance clustering-based dialog models and word embedding techniques. A sentence is described as a vector with the mean value of the constituent word embeddings. K-means clustering is performed using the sentence vector. Here, similar sentences in vector space are located close to each other, while different sentences are located far from each other. The vector space becomes the clustering representation space. One drawback of the model proposed in this paper is that it does not consider the validity of clustering. However, because clustering is applied to learning and evaluation in a limited domain, and response sentences of a limited type are targeted for clustering, it is assumed that the clustering error is insignificant. The proposed dialog approach is illustrated in Figure 2.

First, the dialog dataset is described. In the ith step of a conversation, the history is $\mathcal{H}_i=\{s_{i-n},s_{i-n+1},\ldots,s_i\}$, where s_i is the utterance at time i. Knowledge consists of m sentences with $\mathcal{K}_i=\{k_1,k_2,\ldots,k_m\}$. The response $\mathcal{A}_i$ is s_{i+1}.

The history encoder is a bidirectional multilayer gated recurrent unit (GRU) [39] that converts the unified sequence of $\mathcal{H}_i$ into subword form and receives it at the input layer. The knowledge encoder receives as input $\mathcal{K}_i$ consisting of m sentences determined by $\mathcal{H}_i$ at time i. Here, $\mathcal{K}_i$ is converted into $m'(<m)$ sentences through a filtering process by measuring the similarity with the sequence embedding of $\mathcal{H}_i$. They are integrated, scaled, and combined with the final state of the history encoder and passed through the fully connected (FC) layer to the decoders.

The response decoder consists of a multilayer GRU. Multiple attentions are applied to the history encoder

and knowledge encoder in different decoder layers. The response decoder then performs response generation. For the response class prediction block, the response list of the dialog dataset is converted into the response class by K-means clustering in advance. The response class prediction infers response class $\mathcal{C}(\mathcal{A}_i)$ using the state information transmitted from the encoders. Training the dialog model attempts to maximize (9).

$$\mathcal{L} = \sum_i \log p(\mathcal{A}_i|\mathcal{H}_i, \mathcal{K}_i) + \log p(\mathcal{C}(\mathcal{A}_i)|\mathcal{H}_i, \mathcal{K}_i). \quad (9)$$

5 | EXPERIMENTS

5.1 | Evaluation of subword embedding

This section describes our evaluation of the sentence-model-based subword embedding performance. We used the word similarity, word analogy, and sentence similarity evaluation sets.

5.1.1 | Settings

Evaluation data. The word similarity evaluation set was the released version of the WS353 evaluation set (*WS*) provided in [40] translated into Korean [17].[3] It used the average similarity of two words (353 pairs) assigned on a scale of 0 to 10 by 10 or more evaluators. Word embedding values for the word pairs were expressed as two vectors, where the performance was evaluated by comparing the cosine similarity of the two vectors with the values assigned by the human graders.

The authors of [17] also released the word analogy evaluation set (*WA*), which consisted of 5000 items for evaluating semantic features and 5000 items for evaluating syntactic features. Additionally, the WA set with 10 000 items was the Korean translation of the English word analogy test sets [37]; however, [17] applied a considerable degree of modification to the evaluation set reflecting the characteristics of the Korean language.

We constructed an evaluation set for our sentence similarity (*SS*) experiments. This set was composed of 1000 sentence pairs consisting of 7 to 15 words each. The average on five scales of the values assigned by 10 evaluators was used as the similarity of the sentence pairs. In this study, we compared the similarity assigned to the sentence evaluation set and the cosine similarity of the two sentence embedding vectors.

[3]https://github.com/SungjoonPark/KoreanWordVectors.

TABLE 1 The words and sentences in the training corpus *word_seg* are composed of morphemes as a result of word segmentation and are used for the subword skip-gram model

Dataset	Words (#)	Sentences (#)
Wikipedia	55.8 M	4.6 M
Online news	46.2 M	3.5 M
Online interview	27.2 M	2.2 M
Total	125.8 M	10.3 M
Total *word_seg*	213.7 M	10.3 M
Total *word_seg* + *sent_div*	206.7 M	11.6 M

Note: *sent_div* is the result of separating sentences into 15–30 to extract sentence pairs for the subword skip-thought model and remove short sentences.

Corpus. The training corpus used in the experiment is described in Table 1. The training corpus consisted of Korean Wikipedia,[4] newspaper articles,[5] and broadcast interviews.[6] An effort was made to emulate the training dataset used in [17].

Parameters. The vocabulary used was limited to words with 10 or more occurrences in *word_seg*, and a vocabulary set of 192 445 was constructed. We extracted subwords with syllable lengths of 2–4 in the subword extraction step for single words. A total of 41 451 subwords were constructed by selecting a subword set with 90% coverage of the entire vocabulary set. The SSGM batch size was 400, the negative sample size was 45, and the distance between the target and context vocabularies was maximally 5. Moreover, the SSTM batch size was 32, the negative sample size was 32, and the distance between the target and context sentences was maximally 5. The size of the subword vector was 300 dimensions. We used the Adagrad optimizer and started with an initial learning rate of 1.0. As the learning progressed, exponential decay was performed. Most parameters were selected empirically.

Comparison. The subword embedding model was largely classified into a word-based sentence model (*WSM*; 6) and a *Eojeol*-based sentence model (*ESM*; 8). The training of the subword model was based on the integrated SSGM and SSTM model illustrated in Figure 1. The learning results consisted of three embedding results for the input subword feature set. The first result was Φ_t, which was shared by SSGM and SSTM, while the second two results were Φ_c and Φ_n, which constituted the SSGM and SSTM contexts, respectively. According to the use of

[4]https://dumps.wikimedia.org/kowiki/20181001/.
[5]https://www.chosun.com/, 2013–2015, include all domains.
[6]https://www.cbs.co.kr/radio, http://radio.ytn.co.kr, http://nbbs3.sbs.co.kr.

the embedding results, the experiment was classified into a common subword vector (*CV*; Φ_t) and an integrated subword vector (*IV*; $(\Phi_t + \Phi_c + \Phi_n)/3$). Each had the following sentence model comparison groups:

- NORMAL (*N*): sentence model without position encoding.
- NORMAL SINUSOID PE (*N-SIN*): sentence model with sinusoidal position encoding [9].
- SELF-ATTENTION (*S*): sentence model with self-attention.
- SELF-ATTENTION RELATIVE PE (*S-REL*): sentence model with self-attention and relative position encoding.
- SELF-ATTENTION SINUSOID PE (*S-SIN*): sentence model with self-attention and sinusoidal position encoding.

5.1.2 | Results

The experimental results are presented in Table 2. *SISG* experimental results were taken from a previous study [17]. *SISG*(*ch*) is a model that uses syllable-based features, while *SISG*(*ch* + *jm*) is a model that uses grapheme-based features in addition to syllable-based features. The *FastText*[7] experiment was conducted with the same experimental settings as in this study except for the default hyperparameter. *SISG* and *FastText* used (3) presented in Section 3.1. The *WSM*.*.* and *ESM*.*.* experiments were conducted four times in total, and Table 2 presents the average evaluation results. The reason for this is that word embedding learning was performed based on random context selection, and there was thus variation in the experimental values according to the learning results.

Overall, *SIN* appeared to be inadequate for SSTM. In the *ESM* experiment, both *SIN* and *S-REL* exhibited poor performance, which occurred because *ESM* included a *Eojeol*-based position encoding design. The overlapping use of position encoding thus produced poor performance. In the *WS* evaluation, *ESM.N.IV* exhibited the best performance, which was a slight improvement over the results of similar studies. In the *SS* evaluation, *WSM.N.CV* and *WSM.S-REL.IV* produced good results. In the *WA* evaluation, however, *FastText* performed best. The degradation in the *WA* evaluation was due to unregistered subwords within 90% coverage. Therefore, we conducted experiments using 1-length syllables as an additional feature and 95% subword coverage. As a result of the experiment, (*WSM.N.CV*) exhibited a performance of 48.78, close to that of *FastText*, for *WA*. For *SS*, (*ESM.N.IV*) demonstrated the highest performance of 0.541. However, none of the models performed well on any of the evaluation sets. It was also difficult to detect performance differences between *CV* and *IV*. However, Figure 3 demonstrates that *IV* was more stable in the learning process than *CV*. Overall, Table 2 suggests that *WSM.N* and *ESM.N* were superior to the other model extensions.

Analysis of *WA* results. The *WA* evaluation verified the linear additive properties of the word embedding [41]. The poorer performance of *WSM* and *ESM* than that of *FastText* indicates that *WSM* and *ESM* disrupted the linear additive properties of the existing *FastText* model. This is because *WSM* and *ESM* were configured to reflect additional sentence context information in *FastText*. It can be concluded that integration of the sentence context information of word embeddings negatively affects with the linear additive properties in the vector space. However, the better performance of *WSM* and *ESM* in downstream task evaluation signifies that *WA* evaluation is not appropriate for language understanding tasks.

5.2 | Experiments on dialog model

5.2.1 | Settings

Dialog dataset. A dialog was created for recommending clothing through language interactions between a user and the system. It contained 100 user profiles and 329 TPOs (time, place, occasion). It also contained various functional tags, such as persuasive utterances (EXP_RES), recommendation success (SUCCESS), and failure (FAIL). We assembled a set of 7236 dialogs, which consisted of user utterances (52 599), system utterances (77 392), and clothing recommendations (25 744). Table 3 presents an example of a conversation.

Evaluation. After removing EXP_RES from the entire dialog, we created a total of 48 292 triples consisting of the conversation history, related clothing information, and responses. The reason for removing EXP_RES is that persuasive utterances are composed of too many expressions for clustering. We decided that a new language generation technique is required to handle EXP_RES; thus, we excluded it from the experiment. The responses here refer only to system utterances (*S*) in Table 3. There were a total of 2604 pieces of clothing, and an individual item was represented by a maximum of 47 sentences. The number of words for the encoder and decoder was 11 305 and 1731, respectively. The training set used 43 292 triples, while the evaluation set used the remaining 3000 triples. To create the response class, we

[7]https://github.com/facebookresearch/fastText.

TABLE 2 WordSim (*WS*) and SentSim (*SS*) use the Spearman ρ

Model	WS	WA	SS
SISG(ch)	0.658	-	-
SISG(ch + jm)	0.677	-	-
FastText	0.656	**50.52**	0.423
WSM.N.CV	0.689 (0.645)	45.13 (48.78)	**0.501** (0.485)
WSM.N.IV	0.658 (0.577)	40.80 (43.69)	0.465 (0.528)
WSM.N-SIN.CV	0.640 (0.557)	31.43 (37.59)	0.373 (0.370)
WSM.N-SIN.IV	0.594 (0.523)	39.45 (41.06)	0.404 (0.386)
WSM.S.CV	0.671 (0.573)	44.21 (39.30)	0.489 (0.447)
WSM.S.IV	0.680 (0.610)	40.93 (42.75)	0.465 (0.474)
WSM.S-REL.CV	0.685 (0.651)	43.98 (47.52)	0.484 (0.479)
WSM.S-REL.IV	0.670 (0.636)	40.91 (44.49)	**0.501** (0.521)
WSM.S-SIN.CV	0.622 (0.521)	31.88 (31.25)	0.369 (0.391)
WSM.S-SIN.IV	0.621 (0.533)	39.38 (42.17)	0.456 (0.444)
ESM.N.CV	0.691 (0.683)	43.82 (47.65)	0.500 (0.500)
ESM.N.IV	**0.703** (0.639)	38.12 (39.43)	0.494 (0.541)
ESM.N-SIN.CV	0.554 (0.595)	34.27 (36.84)	0.420 (0.407)
ESM.N-SIN.IV	0.568 (0.254)	36.41 (34.83)	0.407 (0.478)
ESM.S.CV	0.696 (0.408)	44.92 (34.61)	0.500 (0.430)
ESM.S.IV	0.684 (0.320)	38.32 (37.76)	0.482 (0.412)
ESM.S-REL.CV	0.531 (0.472)	39.54 (43.92)	0.457 (0.480)
ESM.S-REL.IV	0.443 (0.553)	36.55 (38.17)	0.442 (0.515)
ESM.S-SIN.CV	0.628 (0.586)	38.96 (40.71)	0.445 (0.487)
ESM.S-SIN.IV	0.618 (0.427)	37.26 (40.41)	0.459 (0.509)

Note: WordAnalogy (*WA*) evaluates the syntactic and semantic sets together and uses accuracy. *SISG* values are taken from [17]. The experiments (*) include 1-length syllables and increase the subword coverage to 95%. In Table 2, bold means the best result in WS, WA, and SS evaluation. Similarly, bold in Table 4 also displays the best result in each evaluation. The correlation between the results of the two experiments is that the dialog experiment in Table 4 is highly correlated with the SentSim (SS) experiment in Table 2. In other words, it means that WSM.S-REL, an optimal word embedding approach for dialog environment, can be found through a simple SentSim (SS) experiment.

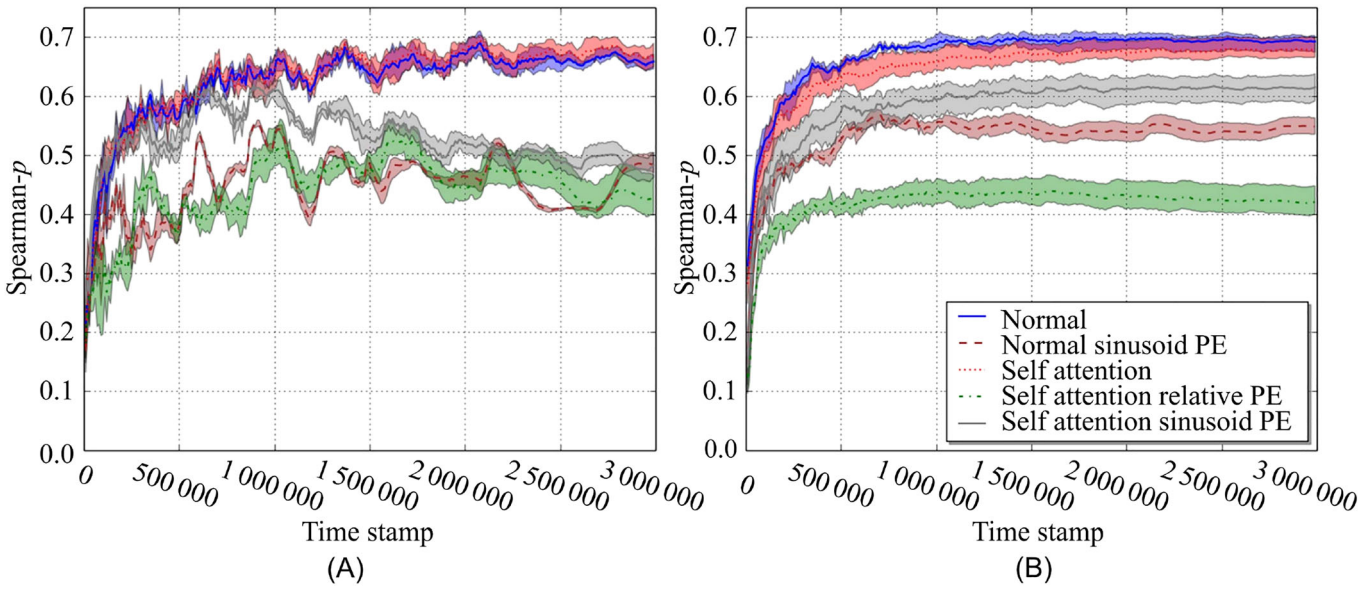

FIGURE 3 WordSim results of (A) *ESM.CV* and (B) *ESM.IV* for different time steps

TABLE 3 Dialog example presenting dialog between the system *S* and a user *U*

	Utterance	Tag
S	Welcome to Codibot. How can I help you?	INTRO
U	Can you show me what clothes I can wear	
	when I'm shooting the wedding?	
S	Can I show you some clothes including a white dress?	SUGGEST
U	Yes.	
SP	JK-030 OP-017 SE-014	
S	I recommend it as a dress that can be coordinated with	EXP_RES
	a bold style with an open shoulder and items that go well with it.	
S	Do you like it?	CONFIRM
U	One piece looks like a vacation look.	FAIL
U	Can you show me another dress.	FAIL

Note: *SP* denotes the recommended clothing IDs. The original dialog was in Korean.

generated a response of 29 399 by removing duplicates from the 48 292 response list. We generated 1000 classes through *K*-means clustering.

Parameters. For response set clustering, PQk-means[8] was used. Each vector was divided into four parts, and each part was 8-bit encoded, resulting in a 32-bit product-quantized (PQ) code. We constructed the dialog model with a three-layer GRU for the encoder and for the decoder. The cell size was 512, and a 300-dimensional vector was used for word embedding. Learning was performed using a batch size of 16, and optimization was performed using the Adam optimizer [42]. The learning rate was set to 0.0001. The maximum sentence length of the knowledge encoder was 47, and the maximum input length of the encoder was 300. For the correct response class prediction, softmax was used after six FC layers.

Comparison. Evaluation was performed using the accuracy of the response class prediction when the encoder–decoder dialog model encoded the input history and knowledge. The evaluation procedure first clustered the response set using the subword embedding results and then determined the training and evaluation data based on the clustering results.

The models to be compared selected good effective from *CV* and *IV*, except for *SIN*, which performed poorly on *WSM* and *ESM*. The selected group performed *K*-means clustering three times, each of which was trained and evaluated. *FastText* was performed in a similar manner. The subword embedding comparison groups were as follows:

- *WSM.N* and *ESM.N*: basic sentence models
- *WSM.S* and *ESM.S*: self-attention sentence models. The input layer of the encoder–decoder model also had a corresponding self-attention structure.
- *WSM.S-REL* and *ESM.S-REL*: self-attention sentence models with relative position encoding. The input layer of the encoder–decoder model also had the same sentence model structure.

Three experiments were conducted for each comparison group according to the word embedding structure and learning parameters in the encoder–decoder model:

- *SW/W*: uses the subword or word embeddings as they are and sets it to "trainable = False" during learning.
- *W-Train*: uses the word embedding structure in the input layer and sets it to "trainable = True" during learning.
- *SW-Train*: uses the subword embedding structure in the input layer and sets it to "trainable = True" during learning.

5.2.2 | Results

The clustering-based dialog model experiments with subword embeddings are described in Table 4. FastText was excluded from the *SW-Train* experiment because it did not provide a list of subwords. Overall, the ESM.S-REL model did not perform well; namely, according to Table 2, it demonstrated poor subword embedding performance. This was because different position encodings were used redundantly. This result suggests that the word embedding approaches significantly affect the performance of the dialog system.

In the *W-Train* experiments, all models except for *ESM.S-REL* demonstrated better performance than FastText. Furthermore, in the *SW/W* experiments, all experiments except for *ESM.N* and *ESM.S-REL* demonstrated better performance than FastText. In particular, *WSM.S-REL* had an accuracy of 4.13% and 4.86% higher than that of FastText in the *SW* and *W-Train* experiments, respectively. The best performance, 58.1%, was achieved by the *WSM.S-REL* model when it was trained by *SW-Train*. The results demonstrate that the experimental deviations were also minimal $(+0.20, -0.20)$. These results imply that a sentence model approach based on self-attention and relative position encoding is

[8] https://github.com/DwangoMediaVillage/pqkmeans.

TABLE 4 Results of clustering-based dialog model response class prediction experiment based on subword embedding models

Model	SW/W	W-Train	SW-Train
FastText	53.77% (+0.63, −0.57)	52.87% (+1.13, −0.67)	-
WSM.N	55.43% (+0.57, −0.43)	54.63% (+1.07, −1.13)	55.47% (+0.73, −1.37)
WSM.S	55.10% (+2.30, −3.30)	55.17% (+2.63, −4.17)	55.23% (+1.97, −3.13)
WSM.S-REL	**57.90%** (+1.00, −0.70)	**57.73%** (+0.47, −0.63)	**58.10%** (+0.20, −0.20)
ESM.N	53.40% (+2.00, −3.50)	53.53% (+2.07, −3.03)	53.57% (+1.83, −3.07)
ESM.S	56.03% (+0.77, −0.43)	55.50% (+0.60, −0.90)	55.07% (+1.43, −2.07)
ESM.S-REL	52.23% (+0.47, −0.73)	52.30% (+1.70, −1.10)	51.93% (+1.47, −0.93)

Note: The input layer structure of the dialog model was composed of a word structure (*W*) and subword structure (*SW*). We also conducted experiments based on whether the embedding parameters reflected learning (*W-Train*, *SW-Train*).

appropriate when multiple sentence sequences, such as dialogs, are required.

The authors of [10] improved transformer self-attention using relative position encoding technology. However, we extended the word embedding part of the RNN's input layer to an SSTM and applied self-attention and relative position encoding. The parameters for relative position encoding, rpe^v and rpe^k, were pre-trained in the subword embedding process. This had the effect of pre-trained relative position encoding parameters with large general text. In summary, the results indicate that our proposed subword-based sentence model using self-attention and relative position encoding is an effective approach for clustering-based dialog models.

As indicated in Table 4, the relative position encoding (*-REL*) experiments require additional learning parameters. When learning parameters of the same size were considered, *ESM.S* exhibited the best performance, 56.03%, in the *SW/W* experiment. This demonstrated the validity of the *ESM* model for executing downstream tasks.

5.3 | Experiments as pre-trained models

5.3.1 | Sequence and token classification

Dataset. We additionally performed evaluations using the Naver Sentiment Movie Corpus (*NSMC*)[9] and Named Entity Recognition (*NER*) dataset.[10] *NSMC* was intended for the sequence classification problem, while the *NER* dataset was intended for the token classification of a sequence. *NSMC* was an internet bulletin board that contained many non-grammatical sentences. This dataset consisted of text samples tagged with positive/negative binary tags for movie review results. It consisted of a training set of 150 000 and a test set consisting of 50 000. The *NER* dataset was assigned named entity tags in Korean word units. All 14 entity names were marked with B and I additional classification tags depending on their location. The *NER* dataset consisted of a training set of 81 000 samples and an evaluation set of 9000 samples.

Models. The learning model for the two evaluation sets used the same three-layer bidirectional GRU as the conversation model. For sentimental classification, an additional FC layer was added to the state value of the model for input, and this was extended to a model for binary classification. The *NER* dataset extended the model to classify tokens by named entity tags by adding an additional FC layer to the output hidden value of the RNN.

The *FastText* and *WSM* models were tested on both evaluation sets as embedding values and compared with an existing pre-trained model, the *Electra*-based approach [43]. The experimental results are presented in Table 5.

Comparison. The experiment used the pre-trained subword embeddings (*WSM*) of the conversational task evaluation. Fine-tuning was performed with the learning data of each task, and evaluation was performed with the test data. That is, the word embedding process for additional learning of *FastText* and *WSM* was not performed using the training corpus of the task. Both *FastText* and *WSM* exhibited similar results in the experiment. On *NSMC*, *FastText* exhibited superior performance, while *WSM* exhibited superior performance on *NER*. This result is due to the noisy text in *NSMC* and the pre-processing of *WSM* without normalizing numbers and English in tokens.

In contrast, the *NER* dataset was composed of grammatically correct sentences, as it is a corpus with *NER* tags assigned to each word. Because of this, the performance of *WSM* seems to be well reflected. Another reason why the performance advantage of *WSM* was not demonstrated by the task is that the

[9] https://github.com/e9t/nsmc.
[10] https://github.com/naver/nlp-challenge.

TABLE 5 Results of sequence classification (*NSMC*) and token classification (*NER*)

Model	NSMC (acc.)	NER (F1)	Parameters (#)
Random	49.65	76.35	17 M
FastText	88.35	79.89	17 M
WSM.N	88.06	80.33	17 M
WSM.S	87.89	78.61	17 M
WSM.S-REL	88.02	79.16	17 M
KoElectra [43]	90.63	88.11	110 M

Note: *NSMC* uses accuracy (acc.), while *NER* uses the F1 measure.

input text itself was dependent on the RNN model. For the conversation task, the sentence embedding part is necessary for processing the metadata of clothing; therefore, the usefulness of the sentence structure model of *WSM* was demonstrated.

Electra exhibited better performance than *BERT* and *GPT* and had an optimal model structure for application to sequence and token classification. For *WSM*, the result was far less than the corresponding performance; however, when considering the number of learning parameters, this approach can be considered effective. There was no significant difference in the fine-tuning speed between *WSM* and *FastText* compared with that of *Electra*. However, the amount of text data required for pre-training and the learning speed required only several hours on a single server.

5.3.2 | Large pre-trained model for dialog task

A pre-trained model with advantages in language processing was selected and applied to the dialog task in this study. Two experiments were conducted: one that predicted the response class by combining the conversation history and knowledge and another that predicted the response class using only the conversation history. Electra [43] was selected as the transformer-based pre-trained model because its performance is superior to that of BERT and GPT. The experimental results are presented in Table 6. The hyperparameters were carried out by referring to the NSMC model.

The response class was clustered using the average of Electra's hidden vector values. For knowledge, the descriptions of clothing were shared by Electra, and class prediction was performed by combining the logit of the Electra conversation history and the logit of knowledge with the weight. The experimental results indicate that adequate performance could not be obtained using a simple sequence prediction model.

TABLE 6 Results of the dialog task

Model	Prediction (acc.)
KoElectra (conversation history)	47.4
KoElectra (conversation history + knowledge)	44.5
WSM.S-REL	58.1

Note: Electra was used for response clustering and response class prediction. The WSM.S-REL results are taken from Table 4 (acc.: accuracy).

5.4 | Description of case study

An embodiment of this study is a clothing recommendation system. First, we must build a dialog set such as that in Table 3. Online chats between a customer and salesperson recommending clothing according to the customer's requirements can be used as learning data. In subword embedding, pre-learning is performed using large-capacity text. Thereafter, subword embedding converts each of the system utterances of the dialog set into a sentence vector representation, performs clustering, and then generates an ID of the system utterance. The dialog model takes the history prior to the system utterance as input and performs supervised learning in the form of predicting the system utterance ID. Here, both learning and evaluation depend on clustering using subword embeddings; therefore, an error in the embedding can affect the entire dialog system. In addition, because the system utterance ID is the ID of the utterance set through clustering, the correct sentence can be estimated once more from the corresponding set. This will be pursued in a future study.

6 | CONCLUSIONS AND FUTURE WORK

In this study, the proposed subword embedding technique using sentence information produced better results than existing techniques on word and sentence similarity

evaluation sets. In particular, we investigated the importance and application of word embedding technology through a clustering-based dialog system. We conducted various experiments to determine the relationship between subword embedding performance and dialog system performance. We found that the subword-based sentence model using self-attention and relative position encoding is a promising approach for clustering-based dialog models. We also conducted a dialog experiment using a large-capacity pre-learning model. We determined that high performance can only be achieved by introducing a more complex and task-appropriate structure. Furthermore, we found that it is difficult to use the large-capacity pre-learning model for a task with structural complexity.

In the future, we plan to review the generality of this study in English. In addition, we plan to apply various subword embedding models to sentence generation.

ACKNOWLEDGMENT

This work was supported by an Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [22ZS1100, Core Technology Research for Self-Improving Integrated Artificial Intelligence System].

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest.

ORCID

Euisok Chung https://orcid.org/0000-0001-5091-2508

REFERENCES

1. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Language models are unsupervised multitask learners*, OpenAI Blog **1** (2019), no. 8, 9. https://www.techbooky.com/wp-content/uploads/2019/02/Better-Language-Models-and-Their-Implications.pdf
2. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint, 2018. https://doi.org/10.48550/arXiv.1810.0480
3. Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, *XLNet: Generalized autoregressive pretraining for language understanding*, arXiv preprint, 2019. https://doi.org/10.48550/arXiv.1906.08237
4. N. Pappas and J. Henderson, *Deep residual output layers for neural language generation*, arXiv preprint, ICML, 2019, pp. 5000-5011. https://doi.org/10.48550/arXiv.1905.05513
5. S. Kumar and Y. Tsvetkov, *Von mises-fisher loss for training sequence to sequence models with continuous outputs* (The Seventh International Conference on Learning Representations, New Orleans, USA), May 2019. https://openreview.net/forum?id=rJlDnoA5Y7
6. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality* (NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems, Red Hook, NY, USA), Dec. 2013, pp. 3111–3119.
7. P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, *Enriching word vectors with subword information*, TACL **5** (2017), 135–146. https://www.aclweb.org/anthology/Q17-1010
8. R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, *Skip-thought vectors* (NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems, Cambridge, MA, USA), Dec. 2015, pp. 3294–3302.
9. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need* (NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA), Dec. 2017, pp. 5998–6008.
10. P. Shaw, J. Uszkoreit, and A. Vaswani, *Self-attention with relative position representations* (Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA), 2018, pp. 464–468. https://doi.org/10.18653/v1/N18-2074
11. Y. Matsui, K. Ogaki, T. Yamasaki, and K. Aizawa, *PQk-means: Billion-scale clustering for product-quantized codes* (MM '17: Proceedings of the 25th ACM international conference on Multimedia, Mountain View, CA, USA), 2017, pp. 1725–1733. https://doi.org/10.1145/3123266.3123430
12. V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain, *Unsupervised word embeddings capture latent knowledge from materials science literature*, Nature **571** (2019), no. 7763, 95–98. https://doi.org/10.1038/s41586-019-1335-8
13. O. Levy and Y. Goldberg, *Dependency-based word embeddings* (Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Maltimore, MD, USA), 2014, pp. 302–308. https://doi.org/10.3115/v1/P14-2050
14. B. Athiwaratkun, A. Wilson, and A. Anandkumar, *Probabilistic fasttext for multi-sense word embeddings* (Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia), 2018, pp. 1–11. https://doi.org/10.18653/v1/P18-1001
15. F. Tian, H. Dai, J. Bian, B. Gao, R. Zhang, E. Chen, and T. Y. Liu, *A probabilistic model for learning multi-prototype word embeddings* (Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland), 2014, pp. 151–160.
16. D. Kiela, C. Wang, and K. Cho, *Dynamic meta-embeddings for improved sentence representations*, (Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium), 2018, pp. 1466–1477. https://doi.org/10.18653/v1/d18-1176
17. S. Park, J. Byun, S. Baek, Y. Cho, and A. Oh, *Subword-level word vector representations for Korean* (Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia), 2018, pp. 2429–2438. https://doi.org/10.18653/v1/P18-1226

18. S. Sasaki, J. Suzuki, and K. Inui, *Subword-based compact reconstruction of word embeddings* (Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA), 2019, pp. 3498–3508. https://doi.org/10.18653/v1/N19-1353
19. B. Heinzerling and M. Strube, *BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages* (Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan), 2018.
20. R. Sennrich, B. Haddow, and A. Birch, *Neural machine translation of rare words with subword units* (Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany), 2016, pp. 1715–1725. https://doi.org/10.18653/v1/P16-1162
21. S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Comput. **9** (1997), no. 8, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
22. T. Kenter, A. Borisov, and M. de Rijke, *Siamese CBOW: Optimizing word embeddings for sentence representations* (Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany), 2016, pp. 941–951. https://doi.org/10.18653/v1/P16-1089
23. E. Chung, H. J. Jeon, S. J. Lee, and J. G. Park, *Korean phoneme sequence based word embedding* (Proceedings of HCLT), 2017, pp. 225–227. http://www.koreascience.or.kr/article/CFKO201731951960129.page
24. H. Chen, X. Liu, D. Yin, and J. Tang, *A survey on dialogue systems: Recent advances and new frontiers*, SIGKDD Explor. Newsl. **19** (2017), no. 2, 25–35. https://doi.org/10.1145/3166054.3166058
25. A. Bordes, Y. Boureau, and J. Weston, *Learning end-to-end goal-oriented dialog* (Proceedings of ICLR), 2017. https://openreview.net/forum?id=S1Bb3D5gg
26. T. H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P. H. Su, S. Ultes, and S. Young, *A network-based end-to-end trainable task-oriented dialogue system* (Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain), 2017, pp. 438–449.
27. Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li, *Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots* (Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada), 2017, pp. 496–505. https://doi.org/10.18653/v1/P17-1046
28. Z. Ji, Z. Lu, and H. Li, *An information retrieval approach to short text conversation*, arXiv preprint, 2014. https://doi.org/10.48550/arXiv.1408.6988
29. R. Yan, Y. Song, and H. Wu, *Learning to respond with deep neural networks for retrieval-based human-computer conversation system* (SIGIR '16: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pisa, Italy), 2016, pp. 55–64. https://doi.org/10.1145/2911451.2911542
30. C. Xing, Y. Wu, W. Wu, Y. Huang, and M. Zhou, *Hierarchical recurrent attention network for response generation* (Proceedings of AAAI), 2018, pp. 5610–5617. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewPaper/16510
31. M. Qiu, F. L. Li, S. Wang, X. Gao, Y. Chen, W. Zhao, H. Chen, J. Huang, and W. Chu, *AliMe chat: A sequence to sequence and rerank based chatbot engine* (Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada), 2017, pp. 498–503. https://doi.org/10.18653/v1/P17-2079
32. Y. Song, R. Yan, X. Li, D. Zhao, and M. Zhang, *Two are better than one: An ensemble of retrieval-and generation-based dialog systems*, arXiv preprint, 2016. https://doi.org/10.48550/arXiv.1610.07149
33. Y. Song, R. Yan, C. T. Li, J. Y. Nie, M. Zhang, and D. Zhao, *An ensemble of retrieval-based and generation-based human-computer conversation systems* (Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence Main track), 2018, pp. 4382–4388. https://doi.org/10.24963/ijcai.2018/609
34. H. Cuayáhuitl, D. Lee, S. Ryu, S. Choi, I. Hwang, and J. Kim, *Deep reinforcement learning for chatbots using clustered actions and human-likeness rewards*, arXiv preprint, IJCNN, 2019, pp. 1–8. https://doi.org/10.48550/arXiv.1908.10331
35. R. C. Gunasekara, D. Nahamoo, L. C. Polymenakos, D. E. Ciaurri, J. Ganhotra, and K. P. Fadnis, *Quantized dialog—A general approach for conversational systems*, Comput. Speech Lang. **54** (2019), 17–30. https://doi.org/10.1016/j.csl.2018.06.003
36. H. Perkins and Y. Yang, *Dialog intent induction with deep multi-view clustering* (Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China), 2019, pp. 4014–4023. https://doi.org/10.18653/v1/D19-1413
37. T. Mikolov, K. Chen, G. Corrado and J. Dean, *Efficient estimation of word representations in vector space*, arXive preprint, 2013. https://doi.org/10.48550/arXiv.1301.3781
38. C. Dyer, *Notes on noise contrastive estimation and negative sampling*, arXive preprint, 2014. https://doi.org/10.48550/arXiv.1410.8251
39. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning phrase representations using RNN encoder–decoder for statistical machine translation* (Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar), 2014. https://doi.org/10.3115/v1/D14-1179
40. L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, *Placing search in context: The concept revisited*, ACM Trans. Inform. Syst. **20** (2002), no. 1, 116–131. https://doi.org/10.1145/503104.503110
41. C. Allen and T. Hospedales, *Analogies explained: Towards understanding word embeddings*, arXive preprint, 2019, pp. 223–231. https://doi.org/10.48550/arXiv.1901.09813
42. D. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXive preprint, ICLR, 2015. https://doi.org/10.48550/arXiv.1412.6980
43. J. Park, *Koelectra: Pretrained electra model for Korean*, 2020. https://github.com/monologg/KoELECTRA

AUTHOR BIOGRAPHIES

Euisok Chung received his B.S. degree in computer science from Soongsil University, Seoul, Republic of Korea, in 1997, and his M.S. degree in computer science from Yonsei University, Seoul, Republic of Korea, in 1999. Since 1999, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea, where he is currently a Principal Member of Engineering Staff. His current research interests include natural language processing, machine learning, and spoken dialog systems.

Hyun Woo Kim received his B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Republic of Korea, in 2001 and 2003, respectively. Since 2003, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea, where he is currently a Principal Member of Engineering Staff. His research areas include speech signal processing, meta-learning, and machine learning.

Hwa Jeon Song received his B.S., M.S., and Ph.D. degrees in electronics engineering from Pusan National University, Republic of Korea, in 1993, 1995, and 2005, respectively. From 1995 to 2001, he was a researcher at Hyundai Motor Company. Since 2010, he has been a principal researcher in Electronics and Telecommunications Research Institute (ETRI). His research interests include speech recognition, multi-modal representation, and artificial general intelligence.

How to cite this article: E. Chung, H. W. Kim, and H. J. Song, *Sentence model based subword embeddings for a dialog system*, ETRI Journal (2022), 1–14. https://doi.org/10.4218/etrij.2020-0245