IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 18, 2022, accepted March 27, 2022, date of publication April 4, 2022, date of current version April 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164769

TA-SBERT: Token Attention Sentence-BERT for Improving Sentence Representation

JAEJIN SEO[1], SANGWON LEE[1], LING LIU[2], (Fellow, IEEE), AND WONIK CHOI[1,3], (Member, IEEE)

[1]Department of Electrical and Computer Engineering, Inha University, Incheon 22212, Republic of Korea
[2]College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA
[3]DeepCardio Company Ltd., Incheon 22212, Republic of Korea

Corresponding author: Wonik Choi (wichoi@inha.ac.kr)

This work was supported in part by the Korea Agency for Infrastructure Technology Advancement (KAIA) Grant funded through the Ministry of Land, Infrastructure and Transport under Grant 22BDAS-C158275-03; and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) Grant funded by the Korea Government (MSIT) (Artificial Intelligence Convergence Research Center, Inha University) under Grant 2020-0-01389.

ABSTRACT A sentence embedding vector can be obtained by connecting a global average pooling (GAP) to a pre-trained language model. The problem of such a sentence embedding vector using a GAP is that it is generated with the same weight for all words appearing in the sentence. We propose a novel sentence embedding-method-based model Token Attention-SentenceBERT (TA-SBERT) to address this problem. The rationale of TA-SBERT is to enhance the performance of sentence embedding by introducing three strategies. First, we convert the base form while preprocessing the input sentence to reduce misunderstanding. Second, we propose a novel Token Attention (TA) technique that distinguishes important words to produce more informative sentence vectors. Third, we increase stability of fine-tuning to avoid catastrophic forgetting by adding a reconstruction loss to the word embedding vector. Extensive ablation studies demonstrate that our TA-SBERT outperforms the original SentenceBERT (SBERT) in the sentence vector evaluation using semantic textual similarity (STS) tasks and the SentEval toolkit.

INDEX TERMS Natural language processing, sentence representation, semantic textual similarity, BERT, RoBERTa.

I. INTRODUCTION

Natural language processing (NLP) is a field of machine learning that enables computers to recognize and analyze human language. The performance of the field varies greatly depending on how the language is vectorized through embedding for natural language data entered by input. Conventional static word embedding such as Word to Vector (Word2Vec) [1], FastText [2], and Global Vectors for Word Representation (GloVe) [3] cannot produce word embedding vectors that reflect the meaning of context. This problem has been solved by the emergence of contextualized word embedding such as Embeddings from Language Models (ELMo) [4], Bidirectional Encoder Representations from Transformers (BERT) [5], Generative Pre-Training (GPT) [6], and Robustly Optimized BERT Pre-Training Approach (RoBERTa) [7], which progresses the supervised learning using the average for the output vectors of BERT, performs efficient sentence embedding.

Transformer [8] is a base model to extract contextualized word embedding. The main idea of Transformer is that multi-head attention is used to process data in parallel while maintaining the time sequence attribute of time-series data by adding positional embedding to the embedding layer.

Unlike the recurrent neural network (RNN), which cannot capture long-term dependencies due to gradient vanishing, Transformer solves this problem by processing input data simultaneously without sequential processing using a self-attention mechanism and a full connected layer. On the other hand, the convolution neural network (CNN) has limited receptive fields due to the kernel size limit, whereas Transformer uses the attention module computing the interaction among all inputs. Recent pre-trained language models (e.g., ELMo, BERT, GPT, and RoBERTa) are trained with large corpora using only the encoder or decoder of

The associate editor coordinating the review of this manuscript and approving it for publication was Ángel F. García-Fernández.

Transformer, dramatically improving the performance of downstream tasks in NLP.

The limitation of a pre-trained language model is that it requires extensive time in sentence pair regression such as clustering and sentence similarity analysis. The problem can be solved by embedding the sentence. Recently, various sentence embedding methods are proposed with different generating mechanisms such as the average of word embedding vectors, special tokens from a pre-trained language model etc. To our knowledge, Sentence-BERT (SBERT) [9], which progresses the supervised learning using the average for the output vectors of BERT, performs efficient sentence embedding.

Even the same word has different meaning in different sentences. Therefore, when interpreting a sentence, the influence of a word in the sentence depends on its role and context. However, the traditional sentence embedding method used in SBERT does not take these characteristics into account because it basically assumes that all words appearing in the sentence have the same weights. To address this issue, we introduce Token Attention-SBERT (TA-SBERT), which produces more informative sentence vector by evaluating higher weighting values for important words. Consequently, our method improves the performance of downstream tasks due to the high quality of sentence vectors.

Our method has three main contributions. First, when an apostrophe (') symbol appears in a sentence, the existing tokenizers merely divide them into token units without considering the base form of the input sentence. In our method, the abbreviations are converted to base form before dividing them into token units and then used as an input to the neural network. Second, when we understand a sentence, we do not interpret the sentence with the same weight on the words in the sentence. Our Token Attention (TA) technique is introduced to reflect these behaviors by finding important words and assigning high weights to important words when generating the sentence vector. Third, BERT and RoBERTa perform a masked-language modeling (MLM) task when pre-training the neural network using large datasets. This MLM task is performed by changing some tokens by randomly selecting using three methods (mask, replace, and maintain) and then matching the input token by adding a linear projection layer to the output vector. However, when only sentence vectors are trained, pre-trained weights change significantly and stability of fine-tuning decreases. We address this issue by adding the reconstruction loss to the word embedding vector to avoid catastrophic forgetting.

II. RELATED WORK

In NLP, the model performance on various tasks depends primarily on how input data are transformed into a vector. In order to vectorize the input data, the tokenization opera-tion is performed first performed. The tokenization includes subword tokenization such as byte pair encoding (BPE) [10], WordPiece encoding [11], the Unigram Language Model [12], and SentencePiece [13].

Subword tokenization solves the Out Of Vocabulary (OOV) problem, which was a problem for tokenizers in the past that generates tokens conveying the meaning of words and can adjust the number of words in the vocabulary. The BPE method divides the text into character units and combines the most frequent combination of characters based on the frequency in a given dataset. In contrast, the WordPiece tokenizer combines characters with the highest likelihood. Even if the input word is not in the vocabulary, tokenization can proceed by combining the word with other tokens in the vocabulary.

In contrast to BPE or WordPiece, the Unigram algorithm defines the loss of training data given to the current vocabulary and Unigram language model. For each symbol in the vocabulary, the algorithm calculates how much the overall loss increases when the symbol is removed from the vocabulary and generates a vocabulary that gradually removes the vocabulary of the current model. The tokenization methods mentioned so far assume that input text uses spaces as a delimiter. In contrast, SentencePiece can train subword models directly from raw sentences without pre-tokenized into word sequences.

The task of encoding the meaning of tokens and changing them to vectors to perform text analysis is referred to as word embedding. Static word embedding methods include techniques such as Word2Vec [1], FastText [2], and GloVe [3]. Word2Vec has two models. Continuous Bag of Words (CBOW) predicts middle words with surrounding words, whereas Skip-Gram predicts surrounding words with middle words. FastText is an extension of Word2Vec and trains by considering that there are several subwords in one word. GloVe trains by reflecting the statistics of the whole corpus and the relationship between surrounding words.

These static word embedding methods store the trained vector and use the stored vector to vectorize input data. Because even the same word has different meanings, there is a problem that the stored vector may not be able to express the corresponding input. This problem can be addressed using a pre-trained language model that extracts contextualized word embeddings, such as ELMo [4], BERT [5], GPT [6], and RoBERTa [7].

The pre-trained language models are constructed using only the encoder or decoder of the transformer and perform pre-training on word embeddings on large corpora. These models can extract a word vector of contextual meaning using a value obtained by performing tokenization and vectorization of input data as the input of a language model. Contextualized word embedding vectors that consider the context of input data outperform static word embedding vectors in various downstream tasks.

Most NLP models extract sentence vectors using word embedding vectors by adding a global average pooling (GAP) to the end of a model. Sentence to Vector (Sent2Vec) [14] extracts a sentence vector using a GAP. The training method of Sent2Vec is the same as that of Word2Vec. The difference between them is that Sent2Vec dynamically adjusts a window

size to fit the length of the sentence instead of static window size to create a training dataset.

Skip-Thought [15] has the RNN model configuration of an encoder-decoder and trains the neural network by generating front and rear sentences from the decoder using the hidden state value of the last layer of the encoder. The final hidden state value from the trained model encoder is used as a sentence vector. InferSent [16] generates sentence vectors through max-pooling of word vectors obtained using the bidirectional long short-term memory (BiLSTM) model. InferSent consists of Siamese networks and uses the NLI dataset (Stanford Natural Language Information data (SNLI) [17] and Multi-Genre NLI data (MNLI) [18]) for training. The framework of natural machine translation with multiple encoders and decoders [19] simultaneously to achieve universal multilingual and modal representations. Multi-task Dual Encoder Training [20] embeds text from 16 languages into a single semantic space using a multi-task trained dual encoder that learns tied representations using translation-based bridge tasks.

Universal Sentence Encoder (USE) [21] converts transformer encoder-output vectors into sentence vectors using mean-pooling. USE is trained with SNLI, Wiki, and News by configuring Siamese networks. Sentence-BERT (SBERT) [9] uses the value obtained through mean-pooling as a sentence vector for word vectors obtained using BERT and trains this model using Siamese networks and NLI data. SBERT-WK [22] resets the weights of the output vector by using the fluctuation trend of word representation for each layer of the SBERT encoder and extracts the final sentence vector through the weighted sum.

CNN-SBERT [23] extracts a sentence vector by adding a CNN module. The input data length has a fixed value to match the matrix dimension of the CNN module, and a pad is added to enforce the same length for all data. CF-SBERT [24] uses the Siamese BERT and extracts the sentence vector using a GAP. A new sentence is generated using important component data obtained using Part-Of-Speech (POS) tagging from input data. When training and inferencing CF-SBERT, the original and generated sentence are grouped and used as input.

This paper demonstrates that performance can be improved without increasing the size of datasets or the number of trainable parameters by preprocessing the input data. Furthermore, an attention module and reconstruction layer are added to SBERT to enable the model to train important words from the input data and avoid catastrophic forgetting.

III. PROPOSED METHOD AND MODEL ARCHITECTURE

A. BASE FORM CONVERSION METHOD

During the tokenization of each language model, even the same input sentence may have different tokens according to the vocabulary of the tokenizer. The tokenizer vocabularies used by the pre-trained BERT and RoBERTa have 30,522 and 50,257 words, respectively. An example of the results

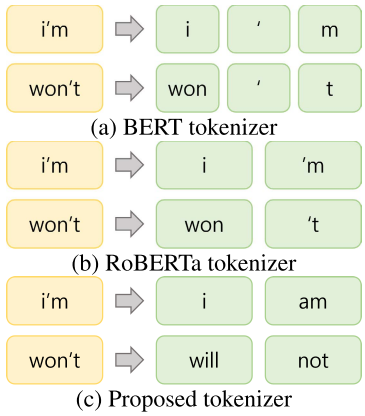


FIGURE 1. Comparison of BERT, RoBERTa and proposed tokenizer.

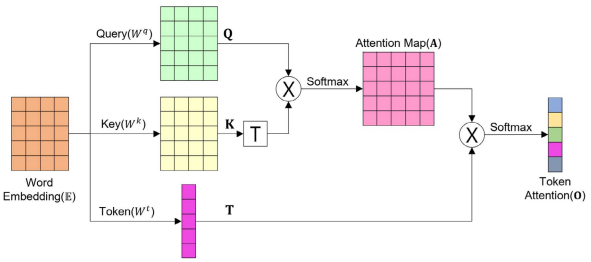


FIGURE 2. Our proposed token attention architecture.

using the BERT and RoBERTa tokenizers is depicted in Fig. 1(a) and 1(b). The BERT tokenizer with a relatively small vocabulary decomposes the apostrophe, whereas the RoBERTa tokenizer does not decompose the apostrophe (').

In contrast, both tokenizers generate a "won" token. Unfortunately, the word "won" has various meanings, such as the past form of "win" and Korean monetary units. When we see the word "won't," we understand it as "will not" intuitively, but a machine cannot. We address this issue by preprocessing the word by converting it to its base form, as depicted in Fig. 1(c).

The base form conversion method can easily remove apostrophes in the input sentence. The apostrophe has various uses. For example, it is used to form possessive nouns and represent the omission of letters. If an input sentence containing abbreviated words using apostrophes is input into a natural language model, this model may not interpret the word as intended. We reduce this uncertainty by removing apostrophes as extensively as with preprocessing using the spaCy Python library, as depicted in Fig. 1(c). Our base form conversion method improves performance by intuitively helping to analyze the contextual meaning of words inside the neural network when fine-tuning is performed (Section IV-B).

B. TOKEN ATTENTION

In interpreting the meaning of a sentence, the importance of every word in the sentence differs. For example, when

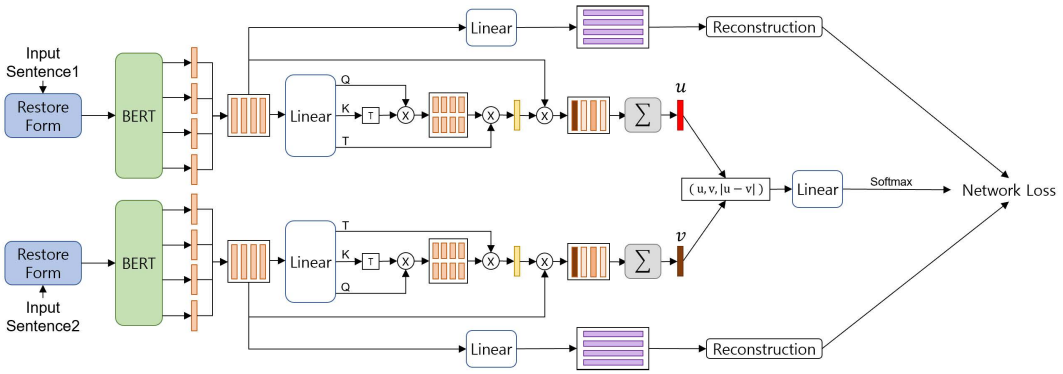


FIGURE 3. Structure of the training model.

a sentence is classified into positive and negative, specific words of affirmation control the overall context of the sentence. The rest of the words are auxiliary in providing additional explanations. Therefore, while generating the sentence vector, it is more reasonable to adjust the weights dynamically than to treat them equally as in a GAP. We propose a novel TA method that finds dynamic weights for important words in the sentence.

Fig. 2 illustrates the proposed TA architecture. The word embedding vector value obtained from the pre-trained language model (Word Embedding Matrix $\mathbb{E} \in \mathbb{R}^{s\times d}$, [$s$ is the sequence length $0 < s \leq 128$, d is the hidden dimension]) is used as the TA input. The query, key, and token matrices are generated from the dot product of matrix $\mathbb{E}$, and each trainable matrix $W^q \in \mathbb{R}^{d\times d}$, $W^k \in \mathbb{R}^{d\times d}$, $W^t \in \mathbb{R}^{1\times d}$.

$$\mathbf{Q} = \mathbb{E} \cdot W_q^T, \quad \mathbf{K} = \mathbb{E} \cdot W_k^T, \ \mathbf{T} = \mathbb{E} \cdot W_t^T \tag{1}$$

In (6), $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{T}$ are query, key, and token ($\mathbf{Q} \in \mathbb{R}^{s\times d}$, $\mathbf{K} \in \mathbb{R}^{s\times d}$). Token has the form of $\mathbf{T} \in \mathbb{R}^{s\times 1}$ and activates only important features for each token.

$$\mathbf{A} = softmax(\frac{\mathbf{QK}^T}{\sqrt{d}}) \tag{2}$$

$$\mathbf{O} = softmax(\frac{\mathbf{AT}}{\sqrt{s_{max}}}) \tag{3}$$

Attention map $\mathbf{A}$ is calculated according to (2). The attention map contains information on how closely the input word embedding vector $\mathbb{E}$ is related to other words and is the same as the single-head attention query and key calculation method. Refer to (3), it calculates the influence of tokens in a sentence. s_{max} in (3) refers to the temperature scaler for TA, which is used to control the variance of $\mathbf{O}$ based on the input sentence length.

$$\mathbf{V} = \sum \mathbb{E} \odot \mathbb{O} \tag{4}$$

The sentence embedding vector $\mathbf{V}$ is calculated as shown in (4) by summing vectors multiplied by element-wise between the influence of the token obtained as a result of TA and the word embedding vector $\mathbb{E}$. In (4), $\mathbb{O}$ means a vector adjusted to match the dimensions of $\mathbf{O}$ in order to perform element-wise multiply with $\mathbb{E}$. Using Token Attention, we can obtain sentence embedding vector weighted according to the importance of the input token.

C. RECONSTRUCTION LOSS FOR AVOIDING CATASTROPHIC FORGETTING

Language models are pre-trained primarily using the MLM task. The task randomly selects some tokens (15%) from among the tokens of the sentence entered as input. Then, 80% of the selected tokens are replaced by [MASK] tokens, 10% by changing other random words, and the remaining 10% is left unchanged. And then a model trains to predict actual IDs of randomly selected tokens.

Our reconstruction loss is constructed differently from the MLM task of the training method. In the MLM task, a random subset of the tokens is masked, and the objective function is used to predict the correct identities of the masked tokens. However, we must consider all words in an input sentence without masking any word to generate the importance of all words. Therefore, the reconstruction loss is set to predict the IDs of all tokens. The purpose of this configured reconstruction loss increases the stability of fine-tuning to avoid catastrophic forgetting problem.

$$\mathbf{r} = softmax(\mathbb{E} \cdot W_r^T) \tag{5}$$

In (5), the linear projection ($W_r \in \mathbb{R}^{C\times d}$) is used to match the word embedding vector $\mathbb{E}$ with the vocabulary of the tokenizer. We use cross-entropy loss to optimize the reconstruction objective function between the input tokens ID and $\mathbf{r}$.

D. TRAINING AND INFERENCE MODEL

SBERT, based on BERT and RoBERTa, has the Siamese and triplet networks. Similar to SBERT, our TA-SBERT has the Siamese network structure, as illustrated in Fig. 3. The proposed TA-SBERT uses the reconstruction loss and has two objective functions: (i) the sentence objective function trains the sentence vector and (ii) the reconstruction objective function trains the reconstruction loss. The sentence objective function is calculated using the sentence vector for the

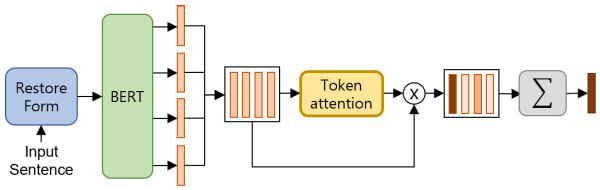


FIGURE 4. Structure of the inference model.

sentence pair as described in (6).

$$\mathbf{s} = softmax((u, v, |u - v|) \cdot W_s^T) \tag{6}$$

In (6), u and v are sentence embedding vectors of the sentence pair data obtained by the pre-trained language model and Token Attention though (1)-(4). We concatenate the sentence embedding vectors u and v with element-wise difference $|u - v|$ and multiply it with trainable weight matrix $W_s \in \mathbb{R}^{k \times 3d}$ to match a dimension of label classes in the dataset. k means the number of label classes in the dataset. We use cross-entropy loss to optimize the sentence objective function between the target labels in the dataset and $\mathbf{s}$ in (6). The reconstruction objective function is calculated as shown in Section III-C

When our model trains at a 1:1 ratio between the sentence objective function and the reconstruction objective function, it tends to overfit the reconstruction task.

$$\begin{aligned}\mathcal{L}_{total} &= \mathcal{L}_{sen}(u, v) \\ &\quad + 0.017 \times (\mathcal{L}_{recon}(\mathbf{r}_1) + \mathcal{L}_{recon}(\mathbf{r}_2))\end{aligned} \tag{7}$$

Therefore, we introduce a temperature scaler to match the convergence rate between the sentence objective function and the reconstruction objective function as shown in (7). $\mathbf{r}_1$ and $\mathbf{r}_2$ are generated by the word embedding vectors of the sentence pair data as described in (5). Based on these observations of our extensive experiments, the optimal ratio of the temperature scaler between the sentence objective function and the reconstruction objective function is 1:0.017.

Fig. 4 illustrates the inference architecture of our proposed model. First, the input sentence is converted to a base form (Section III-A) and a contextualized word embedding vector is obtained through a language model. Second, the contextualized word embedding vector is used so TA can obtain the importance of input words. Finally, the sentence vector is extracted based on (4) using both the contextualized word embedding vector and the importance from the TA.

IV. EXPERIMENT

In this section, we analyze the efficacy of our three main techniques to improve the performance of the sentence vector. We evaluate the performance of TA-SBERT for semantic textual similarity (STS) and classification tasks. For evaluation of STS tasks, we use the cosine similarity to compare the similarity between two sentence vectors. The Pearson's and Spearman's rank correlation coefficients are calculated between the cosine similarity of the sentence vectors and gold labels. For classification tasks, we use the SentEval [25] toolkit for measuring the quality of the sentence vector. We use pre-trained BERT and RoBERTa from Hugging Face[1] [26]

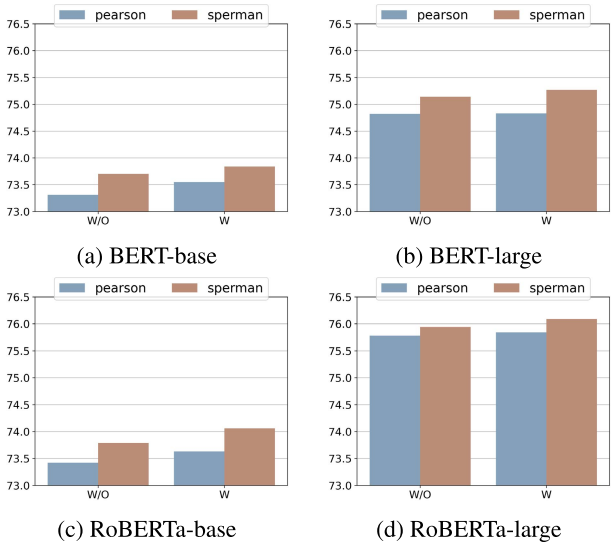


FIGURE 5. The fine-tuning uses only base form conversion method. Pearson correlation coefficient and Spearman's rank correlation coefficient are expressed by multiplying the value by x100 between the cosine similarity of sentence vectors and the gold labels. W/O: without conversion, W: with conversion.

A. TRAINING DETAILS

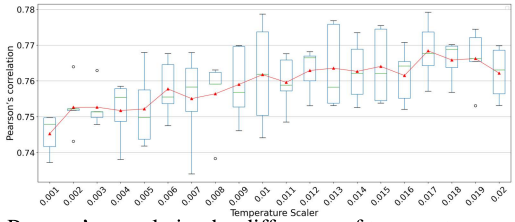
In all experiments, we train TA-SBERT using NLI dataset combined SNLI with MultiNLI. The SNLI is a collection of 570k sentence pairs labeled as entailment, contradiction, and neutral. The MultiNLI corpus is a collection of 433k sentence pairs annotated with textual entailment information. we fine-tune TA-SBERT with one sentence objective function using the NLI dataset and two reconstruction objective functions. We train all our models using a batch size of 16, an epoch of 1, and the AdamW optimizer with a linear learning rate warm-up of 10% of the training data. We use a learning rate of 2e-5 for BERT and RoBERTa as pre-trained language models and 3e-5 for TA and other parameters. All parameters of TA are initialized to a uniform distribution with a mean of 0 and a variance of 0.02.

B. BASE FORM CONVERSION EFFECT

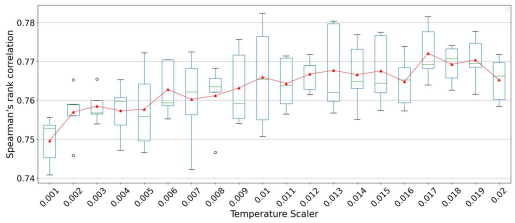
The first set of experiments is designed to evaluate the effectiveness of converting input words to their base forms. This experiment is conducted by fine-tuning the pre-trained BERT and RoBERTa. As presented in Table 3, the result of this experiment shows that the basic form conversion method does not contribute to performance improvement if it is used alone.

In contrast, the use of both TA and base form conversion methods produce meaningful effects, as depicted in Fig. 5, using the STS benchmark (STS-B) [27] dataset. The average performance is improved in all four models. The

[1] https://huggingface.co/



(a) Pearson's correlation by difference of temperature scaler



(b) Spearman's rank correlation by difference of temperature scaler

FIGURE 6. Change of Pearson/Spearman value according to temperature scaler between sentence objective function and reconstruction objective function.

Pearson's and Spearman's rank correlation coefficients average increases of 0.15 points and 0.175 points, respectively, when applying the base form conversion method to the input sentence. This result demonstrates that the base form conversion method is not effective when applied only to a pre-trained language model but is effective in training TA.

C. TOKEN ATTENTION VISUALIZATION

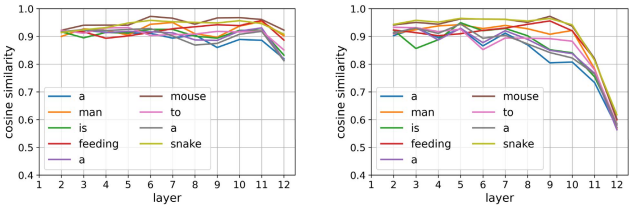
We design our TA technique to focus on important words in the sentence. We illustrate the effectiveness of our TA module by visualizing its sentence output from several different datasets using BERT-base-uncased and the base form conversion, as presented in Tables 1 and 2.

Table 1 presents the output of the trained TA in a model without reconstruction loss. The TA module emphasizes more informative words. In contrast, Table 2 presents the output of the trained TA with the reconstruction loss. The TA module with the reconstruction loss focuses not only on important words but also on words influenced by surrounding words.

D. FIND THE APPROPRIATE LEARNING RATIO BETWEEN SENTENCE LOSS AND RECONSTRUCTION LOSS

In this section, we conduct experiments for finding the optimal temperature scaler between the sentence objective function and reconstruction objective function. TA-SBERT is trained using the NLI dataset. We evaluate our model on the STS-B dataset. The experiment is conducted by setting the temperature scaler of the reconstruction loss from 0.001 to 0.02.

Fig. 6 illustrates the experimental results obtained under these settings, and the red line denotes the average value obtained in each case. The x-axis represents the value of the temperature scaler of the reconstruction loss, while the y-axis of Fig. 6(a) represents the Pearson's correlation coefficient, and the y-axis of Fig. 6(b) represents the Spearman's rank correlation coefficient.



(a) BERT (b) SBERT

FIGURE 7. Cosine similarity of word representation by layer of encoder using BERT and SBERT.

The result of the experiment without using reconstruction loss is described in Fig. 5(a). The values of the Pearson's and Spearman's rank correlation coefficients are 73.31 and 73.7, respectively when the base form conversion method is not used. These values increase to 73.55 and 73.84, respectively, when the base form conversion method is used, as depicted in Fig. 5(a). In contrast, if we add a reconstruction loss, the values of the Pearson's and Spearman's rank correlation coefficients increase to 76.81 and 77.18, respectively, as depicted in Fig. 6.

In this temperature scaler experiment, the temperature scaler of the reconstruction loss performs optimally when the value of the temperature scaler is at 0.017. Based on this result, we set the temperature scaler between the sentence objective function and the reconstruction objective function to 0.017 as shown in (7).

E. EVOLVING WORD REPRESENTATION

We verify the efficacy of the reconstruction loss in word representation from the encoder layers in Figs. 7-8. Figs. 7-8 illustrate how much word representations changes in terms of the cosine similarity between hidden states of encoder layers. Figs. 7-8 are the result of obtaining the word representation in the sentence "A man is feeding a mouse to a snake." The x-axis represents the encoder layer of models. The y-axis represents the cosine similarity between the current and previous layers.

Fig. 7(a) illustrates that the word representation changes slightly inside when using the pre-trained BERT-base-uncased. Furthermore, Fig. 7(b) is the result of the word representation when training only the sentence vector using SBERT. The word representation is maintained up to the eighth encoder layer and then decreases rapidly. Consequently, the existing parameter weights change drastically while training the sentence vector.

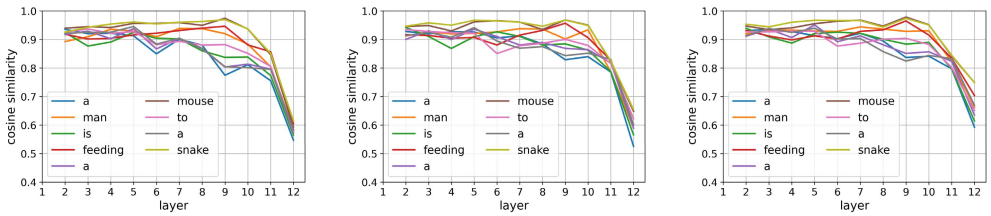
Fig. 8(a) reveals that SBERT with the base form conversion method has similar results as Fig. 7(b). If we add TA to this model, the cosine similarity increases for important words, whereas it decreases for other words, as depicted in Fig. 8(b). Moreover, as depicted in Fig. 8(c), when the reconstruction loss is added to the model in Fig. 8(b), the cosine similarity is highest than other models. In particular, the cosine similarity

TABLE 1. Token attention result without reconstruction loss.

Attention Map	Dataset
this thing plays everything i feed it	CR
worth the effort to watch	MR
a man is feeding a mouse to a snake	STS
whistle completes the trilogy started in from here to eternity and the thin red line	SUBJ

TABLE 2. Token attention result with reconstruction loss.

Attention Map	Dataset
this thing plays everything i feed it	CR
worth the effort to watch	MR
a man is feeding a mouse to a snake	STS
whistle completes the trilogy started in from here to eternity and the thin red line	SUBJ



(a) SBERT+form (b) SBERT+form+att (c) SBERT+form+att+recon

FIGURE 8. Cosine similarity of word representation by layer of encoder fine-tuned models with our proposed methods.

of "snake", "feeding", and "mouse", which are important words in interpreting the input sentence, is increased. The results show that the model trained by using our methods activates for important words.

F. CATASTROPHIC FORGETTING

Catastrophic forgetting [28] means the tendency of a neural network to forget pre-trained knowledge upon training new knowledge. Many work propose various methods to avoid catastrophic forgetting: STILTs [29] adds supervised learning before fine-tuning to avoid model overfitting and catastrophic forgetting. BERT's text classification fine-tuning method [30] adjusts the learning rate to avoid catastrophic forgetting. Mixout [31] introduces a new regularization that prevents catastrophic forgetting by increasing the stability of fine-tuning. To show the effect of reconstruction loss on catastrophic forgetting, we experiment with the stability of fine-tuning at various learning rates based on the SBERT model. We perform fine-tuning of SBERT with reconstruction loss or without reconstruction loss using NLI's train dataset and use NLI's test dataset for evaluation. The hyper-parameter settings are the same as in Section IV-A, except that only the learning rate changes. A fine-tuning is performed 5 times with different random seeds for each learning rate. Fig. 9 shows cross-entropy loss and accuracy according to various learning rates. In Fig. 9(a) and (b), a solid line represents train loss, and a dotted line represents test loss. In Fig. 9(c) and (d), a red line indicates average accuracy. Even if the learning rate increases, the stability of the fine-tuning with reconstruction loss is maintained, but the stability of the fine-tuning without reconstruction loss becomes unstable. Based on these results, the reconstruction loss has the effect of preventing catastrophic forgetting by increasing the stability of fine-tuning.

G. UNSUPERVISED STS TASKS

We evaluate the performance of the STS tasks using fine-tuning TA-SBERT on the NLI training dataset. The datasets we used to evaluate are the STS tasks 2012- 2016 [32]–[36], the STS benchmark, and the SICKRelatedness [37] dataset. These datasets represent the semantic relevance of the sentence pair as a value between 0 and 5. We determine

TABLE 3. Pearson correlation coefficient (left) and Spearman's rank correlation coefficient (right) are expressed by multiplying the value by x100 between the value obtained through the cosine similarity between sentence vectors obtained using the learned model and the value of the gold labels. The higher the value, the higher the correlation, and it means that the prediction label value obtained using the sentence vector obtained by the model and the gold label value are similar. STS12-STS16: SemEval 2012-2016, STSb: STSbenchmark, SICK-R: SICK Relatedness dataset.

Model	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg
SBERT-NLI-base	64.32/65.61	67.52/70.39	72.25/71.8	75.64/76.8	70.18/75.15	74.71/75.15	83.9/79.44	72.64/73.14
+ form	64.24/65.57	67.47/70.36	72.25/71.86	75.56/76.75	70.36/73.21	74.72/75.34	83.76/79.22	72.62/73.19
+ att	64.77/65.61	68.66/71.34	73.24/72.48	76.22/77.32	71.06/73.79	75.07/75.55	84.2/79.79	73.31/73.7
+ form + att	65.11/65.71	68.97/71.62	73.49/72.81	76.52/77.6	71.25/73.72	74.97/75.46	84.58/79.96	73.56/73.84
+ att + recon	65.88/66.9	69.92/72.6	74.27/73.57	**77.36/78.37**	71.36/73.99	76.8/77.06	**85.48**/80.48	74.44/74.71
+ form + att + recon	**66.27/67.08**	**70.78/73.12**	**74.5/73.73**	77.2/78.19	**71.54/74.32**	**76.81/77.18**	85.46/**80.52**	**74.65/74.88**
SBERT-NLI-large	66.61/67.37	70.88/73.17	74.72/74.16	77.41/78.48	72.1/74.77	75.53/76.07	84.13/80.02	74.48/74.86
+ form	66.51/67.34	69.95/72.41	74.07/73.58	77.39/78.55	72.23/75.08	75.88/76.28	84.29/80.08	74.33/74.76
+ att	66.62/67.75	71.39/73.57	74.68/73.98	77.79/78.76	72.44/74.86	76.3/76.66	84.58/80.44	74.83/75.14
+ form + att	66.46/67.73	71.27/73.7	74.85/74.08	77.67/78.77	72.52/75.14	76.35/76.96	84.69/80.47	74.83/75.27
+ att + recon	67.35/68.04	72.78/75.02	76.15/75.39	78.68/79.51	72.77/75.35	76.32/77.04	85.39/81.21	75.63/75.94
+ form + att + recon	**67.81/68.47**	**73.25/75.53**	**76.6/75.65**	**78.71/79.66**	**73.69/76.06**	**76.73/77.37**	**85.62/81.3**	**76.05/76.29**
SRoBERTa-NLI-base	64.43/65.12	68.58/71.32	72.23/71.87	75.91/76.99	71.58/74.5	73.58/74.15	83.49/79.26	72.83/73.31
+ form	64.72/65.57	68.72/71.67	72.93/72.61	76.42/77.51	71.83/74.7	73.93/74.54	83.64/79.36	73.17/73.71
+ att	64.26/65.21	69.56/72.02	73.1/72.6	76.63/77.59	72.2/74.97	74.33/74.8	83.87/79.33	73.42/73.79
+ form + att	64.47/65.4	69.47/72.01	73.4/72.97	77.3/78.26	**72.38**/75.14	74.56/75.17	83.8/79.5	73.63/74.06
+ att + recon	65.59/66.01	**70.32**/72.61	73.59/73.09	76.98/77.79	71.98/74.79	74.47/75.07	84.54/80.2	73.92/74.22
+ form + att + recon	**65.7/66.55**	70.12/**72.62**	**73.82/73.16**	**77.03/77.89**	72.3/**75.17**	**75.72/76.1**	**84.87/80.39**	**74.22/74.55**
SRoBERTa-NLI-large	67.21/67.72	72.64/74.91	75.58/74.61	79.12/79.84	74.92/77.25	75.2/75.84	83.31/79.47	75.43/75.67
+ form	67.16/67.59	72.43/74.71	75.5/74.88	79.27/80.05	75.22/77.48	74.6/75.25	83.25/79.46	75.35/75.63
+ att	67.28/67.85	73.0/74.91	75.72/74.71	79.28/79.99	75.29/77.59	75.88/76.49	83.97/80.01	75.77/75.93
+ form + att	67.11/67.86	73.22/75.3	76.22/75.41	79.61/80.31	75.3/77.58	75.47/76.06	83.94/80.12	75.84/76.09
+ att + recon	67.58/68.38	**73.82/75.95**	76.63/75.86	79.86/80.65	75.44/77.84	75.69/76.27	84.54/80.51	76.22/76.49
+ form + att + recon	**67.62/68.54**	73.47/75.8	**76.66/75.9**	**80.19/80.98**	**76.14/78.44**	**76.15/76.84**	**85.09/80.84**	**76.47/76.76**

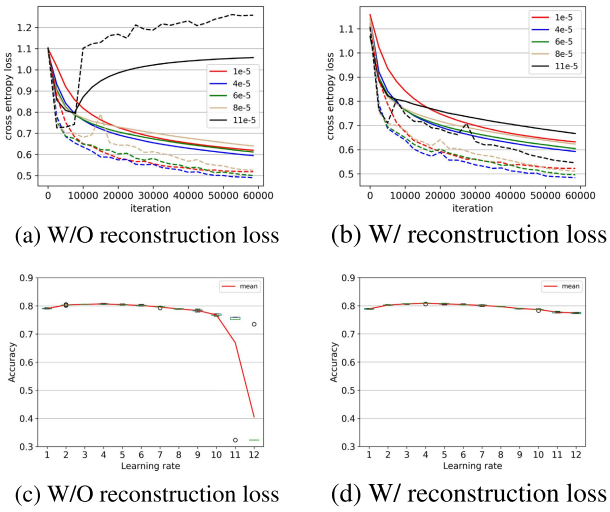
TABLE 4. It is the result of comparing the model used in Table 3 by using the SentEval toolkit. The SentEval toolkit is used to measure the quality of sentence vectors, and the results are measured through a total of 10-fold cross-validation, and the values indicate the accuracy.

Model	MR	CR	MPQA	SUBJ	SST	TREC	MRPC	Avg
SBERT-NLI-base	82.46	**89.26**	89.69	93.13	**88.66**	85.48	75.68	86.34
+ form	82.49	89.16	89.61	93.27	88.24	85.16	**76.15**	86.3
+ att	82.47	89.18	89.73	93.38	88.41	87.08	76.0	86.61
+ form + att	**82.73**	89.2	89.8	93.23	88.11	85.96	76.05	86.44
+ att + recon	82.55	89.16	89.73	93.59	88.49	86.52	75.95	86.57
+ form + att + recon	82.57	89.18	**89.99**	**93.88**	88.31	**87.64**	75.57	**86.73**
SBERT-NLI-large	84.2	90.65	89.98	93.89	90.49	87.84	75.8	87.55
+ form	84.16	90.63	90.09	93.84	90.28	87.8	76.38	87.6
+ att	84.41	90.65	89.9	93.91	**90.69**	88.72	76.21	87.78
+ form + att	84.44	90.53	90.06	**93.97**	90.53	87.7	75.9	87.59
+ att + recon	84.52	**90.74**	90.11	93.96	90.4	89.08	76.12	87.85
+ form + att + recon	**84.53**	90.43	**90.14**	93.93	90.19	**90.32**	**76.76**	**88.05**
SRoBERTa-NLI-base	85.02	**91.52**	89.48	92.87	91.76	85.76	76.44	87.55
+ form	85.22	91.43	89.35	92.84	91.85	86.92	76.76	87.77
+ att	84.97	**91.52**	89.26	93.2	91.76	86.84	77.1	87.81
+ form + att	**85.54**	91.48	89.23	93.23	**92.16**	87.32	76.72	87.95
+ att + recon	85.01	91.18	89.4	**93.55**	91.52	87.04	**77.47**	87.88
+ form + att + recon	84.95	91.47	**89.52**	93.52	91.16	**88.72**	76.95	**88.04**
SRoBERTa-NLI-large	87.0	92.26	90.37	93.94	92.23	90.24	75.77	88.83
+ form	87.15	92.14	90.34	93.89	92.78	91.08	76.0	89.05
+ att	86.89	92.06	90.26	93.88	92.58	90.56	76.17	88.92
+ form + att	**87.27**	92.09	90.33	93.78	**92.93**	90.3	76.74	89.06
+ att + recon	86.75	**92.27**	90.25	94.17	92.74	**91.64**	76.68	89.21
+ form + att + recon	86.86	92.26	**90.52**	**94.32**	92.32	91.04	**77.7**	**89.29**

the similarity between sentence vectors using the cosine similarity of the vector of the input sentence pair obtained using TA-SBERT.

The results of the Pearson's and Spearman's rank correlation coefficients between sentence vectors and gold labels are presented in Table 1 the sentence vector generated by TA-SBERT is superior to SBERT for all datasets. In Table 3, ''+form'' indicates the base form conversion method, ''+att'' the TA, and ''+recon'' the reconstruction loss. The base form conversion method(+from) does not affect performance (Section IV-B). However, when our base form conversion method is used with the TA technique, the Pearson's and Spearman's rank correlation coefficients increase on average by 0.62 and 0.57, respectively, compared with SBERT.

Moreover, when the reconstruction loss is additionally used, the Pearson's and Spearman's rank correlation coefficients increase on average by 1.5 and 1.38, respectively, compared with SBERT. Consequently, the three proposed



(a) W/O reconstruction loss (b) W/ reconstruction loss

(c) W/O reconstruction loss (d) W/ reconstruction loss

FIGURE 9. (a) and (b) show cross-entropy loss during training. (c) and (d) show the accuracy, where the x-axis is the learning rate multiplied by 10000.

techniques the base form conversion method, TA, and the reconstruction loss are significant in generating the sentence vector by reducing the uncertainty of abbreviations, focusing on important words, and avoiding catastrophic forgetting.

The BERT-base-uncased has 768 dimensions of word vectors and 12 encoder layers, whereas the BERT-large-uncased has 1,024 dimensions of word vectors and 24 encoder layers. Even in these settings, the sentence vector generated by BERT-base-uncased with our proposed techniques illustrates similar performance as the sentence vector generated by BERT-large-uncased in the STS task.

H. SENTENCE VECTOR PERFORMANCE

SentEval is a toolkit used to evaluate the quality of sentence vectors. It uses sentence vectors as features to train logistic regression classifiers. When training the logistic regression classifier, we set all parameters to default and evaluate using the following seven datasets:

- MR: Movie Review dataset for the sentiment classification task. [38]
- CR: Customers Reviews dataset for the sentiment classification task. [39]
- SUBJ: Subjectivity prediction from movie reviews. [40]
- MPQA: Phase-level opinion polarity dataset for the sentiment classification task. [41]
- SST: Stanford Sentiment Treebank dataset for the sentiment classification task. [42]
- TREC: Question-type dataset for the multi-class classification task. [43]
- MRPC: Microsoft Research Paraphrase Corpus from newswire articles for the binary classification task. [44]

Table 4 presents the accuracies of the logistic regression classifier using these seven datasets, demonstrating that our TA-SBERT exhibits the highest average accuracy.

V. CONCLUSION

We propose a novel sentence embedding-method-based model, TA-SBERT, with three original contributions. First, we preprocess the input data by converting the word to the base form and removing apostrophes to reduce the uncertainty of the word's meaning. Second, we introduce the TA method to assign weights dynamically to important words in a sentence. Third, we propose the reconstruction loss to avoid catastrophic forgetting by increasing stability of fine-tuning.

We conduct experiments on STS tasks and classification tasks for seven datasets using TA-SBERT. TA-SBERT exhibits average increases of 2.04% and 1.86% for the Pearson's and Spearman's rank correlation coefficients in STS tasks compared with SBERT. In classification tasks using the SentEval toolkit, the average accuracy increases by 0.53%.

As future work, we plan to apply the Token Attention and the reconstruction loss to other downstream tasks including the task of summarizing using sentence vectors or the task of rating each sentence by importance in a long document.

REFERENCES

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Jun. 2016.

[3] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.

[4] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. ACL*, 2019, pp. 4171–4186.

[6] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," Tech. Rep., 2018.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3980–3990.

[10] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1715–1725.

[11] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, and J. Klingner "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.

[12] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 66–75.

[13] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. EMNLP*, 2018, pp. 66–71.

[14] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 528–540.

[15] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3294–3302.

[16] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 670–680.
[17] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 632–642.
[18] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 1112–1122.
[19] H. Schwenk and M. Douze, "Learning joint multilingual sentence representations with neural machine translation," in *Proc. 2nd Workshop Represent. Learn. (NLP)*, 2017, pp. 157–167.
[20] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y. H. Sung, and B. Strope, "Multilingual universal sentence encoder for semantic retrieval," in *Proc. ACL*, Jul. 2020, pp. 87–94.
[21] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," 2018, *arXiv:1803.11175*.
[22] B. Wang and C.-C. Jay Kuo, "SBERT-WK: A sentence embedding method by dissecting BERT-based word models," 2020, *arXiv:2002.06652*.
[23] H. Choi, J. Kim, S. Joe, and Y. Gwon, "Evaluation of bert and Albert sentence embedding performance on downstream nlp tasks," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, 2021, pp. 5482–5487.
[24] X. Yin, W. Zhang, W. Zhu, S. Liu, and T. Yao, "Improving sentence representations via component focusing," *Appl. Sci.*, vol. 10, no. 3, p. 958, Feb. 2020.
[25] A. Conneau and D. Kiela, "Senteval: An evaluation toolkit for universal sentence representations," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, 2018.
[26] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Davison, "Transformers: State-of-the-art natural language processing," in *Proc. EMNLP*, 2020, pp. 38–45.
[27] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 1–14.
[28] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivat.*, vol. 24, pp. 109–165, Dec. 1989.
[29] J. Phang, T. Févry, and S. R. Bowman, "Sentence encoders on STILTs: Supplementary training on intermediate labeled-data tasks," 2018, *arXiv:1811.01088*.
[30] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *Proc. China Nat. Conf. Chin. Comput. Linguistics*. Springer, 2019, pp. 194–206.
[31] C. Lee, K. Cho, and W. Kang, "Mixout: Effective regularization to finetune large-scale pretrained language models," 2019, *arXiv:1909.11299*.
[32] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre, "Semeval-2012 task 6: A pilot on semantic textual similarity," in *Proc. 1st Joint Conf. Lexical Comput. Semantics*, vol. 2, 2012, pp. 385–393.
[33] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, "SEM 2013 shared task: Semantic textual similarity," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics (SEM)*, 2013, pp. 32–43.
[34] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe, "Semeval-2014 task 10: Multilingual semantic textual similarity," in *Proc. 8th Int. workshop semantic Eval. (SemEval)*, 2014, pp. 81–91.
[35] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, G. Rigau, L. Uria, and J. Wiebe, "SemEval-2015 task 2: Semantic textual similarity, english, Spanish and pilot on interpretability," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 252–263.
[36] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez Agirre, R. Mihalcea, G. Rigau Claramunt, and J. Wiebe, "Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *Proc. 10th Int. Workshop Semantic Evaluation (SemEval)*, San Diego, CA, USA, Jun. 2016, pp. 497–511.
[37] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli, "A sick cure for the evaluation of compositional distributional semantic models," in *Proc. 9th Int. Conf. Lang. Resour. Eval. (LREC)*, Reykjavik, Iceland, 2014, pp. 216–223.
[38] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2005, pp. 115–124.
[39] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2004, pp. 168–177.
[40] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2004, p. 271.
[41] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Lang. Resour. Eval.*, vol. 39, no. 2, pp. 165–210, May 2005.
[42] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*, 2013, pp. 1631–1642.
[43] X. Li and D. Roth, "Learning question classifiers," in *Proc. 19th Int. Conf. Comput. Linguistics*, 2002.
[44] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *Proc. 20th Int. Conf. Comput. Linguistics (COLING)*, 2004, pp. 350–356.

JAEJIN SEO received the B.S. degree in mechanical engineering from Inha University, Incheon, Republic of Korea, in 2013, where he is currently pursuing the M.S. degree in electrical and computer engineering under the guidance of Prof. Choi. His research interests include deep learning, natural language processing, recommendation systems, and big data.

SANGWON LEE received the B.S. degree in mechanical engineering from Inha University, Incheon, Republic of Korea, in 2013, where he is currently pursuing the M.S. degree in electrical and computer engineering under the guidance of Prof. Choi. His research interests include deep learning, natural language processing, recommendation systems, and big data.

LING LIU (Fellow, IEEE) is currently a Professor with the School of Computer Science, Georgia Institute of Technology. She directs the research programs with the Distributed Data Intensive Systems Laboratory (DiSL). She has published over 300 international journals and conference papers. She was a recipient of the IEEE Computer Society Technical Achievement Award in 2012 and the best paper awards from numerous top venues. She served as the EIC of the IEEE TRANSACTIONS ON SERVICE COMPUTING (2013–2016). She serves on the editorial board for half a dozen international journals.

WONIK CHOI (Member, IEEE) received the Ph.D. degree in computer engineering from Seoul National University, South Korea. He was a Visiting Scholar with the School of Computer Science, Georgia Institute of Technology, in 2012. He is currently a Professor with the School of Information and Communication Engineering, Inha University, where he runs the Data Intelligence Laboratory. His research interests include spatio-temporal databases, sensor network topology, telematics, and GIS/LBS.