Check for updates

Received: 15 July 2020 | Revised: 30 December 2020 | Accepted: 22 January 2021

DOI: 10.4218/etrij.2020-0282

ETRI Journal WILEY

ORIGINAL ARTICLE

Simple and effective neural coreference resolution for Korean language

Cheoneum Park[1] | **Joonho Lim**[2] | **Jihee Ryu**[2] | **Hyunki Kim**[2] | **Changki Lee**[3]

[1]AIRS Company, Hyundai Motor Group, Seoul, Repulic of Korea

[2]SW and Contents Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon, Repulic of Korea

[3]Computer Science, Kangwon National University, Chuncheon, Repulic of Korea

Correspondence
Changki Lee, Computer Science, Kangwon National University, Chuncheon, Repulic of Korea.
Email: leeck@kangwon.ac.kr

Funding information
Korean Government (MSIT), Grant/Award Number: 2013-0-00131; Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services,

Abstract

We propose an end-to-end neural coreference resolution for the Korean language that uses an attention mechanism to point to the same entity. Because Korean is a head-final language, we focused on a method that uses a pointer network based on the head. The key idea is to consider all nouns in the document as candidates based on the head-final characteristics of the Korean language and learn distributions over the referenced entity positions for each noun. Given the recent success of applications using bidirectional encoder representation from transformer (BERT) in natural language-processing tasks, we employed BERT in the proposed model to create word representations based on contextual information. The experimental results indicated that the proposed model achieved state-of-the-art performance in Korean language coreference resolution.

KEYWORDS

coreference resolution, head-final language, Korean, pretrained language model, recurrent neural network

1 | INTRODUCTION

Coreference resolution [1–3] links words that point to the same entity to a single cluster in a given context. When an entity is represented by another word, there exist aliases, abbreviations, pronouns, deterministic phrases, and other languages. By correctly defining the reference relationship among them, information about the referred entity in the discourse or document can be consistently maintained and accurately conveyed. To perform coreference resolution, each noun in a context is defined as a mention that includes modifiers, where the noun represents a head word. Because Korean is a head-final language [4]; the final word of a mention is always the head; therefore, we can (a) easily obtain the head in Korean, (b) perform coreference resolution based on the head of the mention, and (c) solve mention detection simultaneously by obtaining the start boundary of mentions. In this study, we performed coreference resolution using an end-to-end neural model and an attention mechanism [5,6]. The method using the attention mechanism models mentions in the document to point to the entity position containing the mentions. In this case, the number of solved entities for a coreference resolution is defined as the position of the first mention of each entity. For the example in Figure 1, we present an Entity that is referred with the Heads that point to the same entity. Heads are extracted from the input document (Doc) and referred words among the Heads with Entity point to the first mention, “"종대o(Sejong)”, of the entity and words that do not refer to the dummy symbol, ϵ.

This is an Open Access article distributed under the term of Korea Open Government License (KOGL) Type 4: Source Indication + Commercial Use Prohibition + Change Prohibition (http://www.kogl.or.kr/info/licenseTypeEn.do).
1225-6463/$ © 2021 ETRI

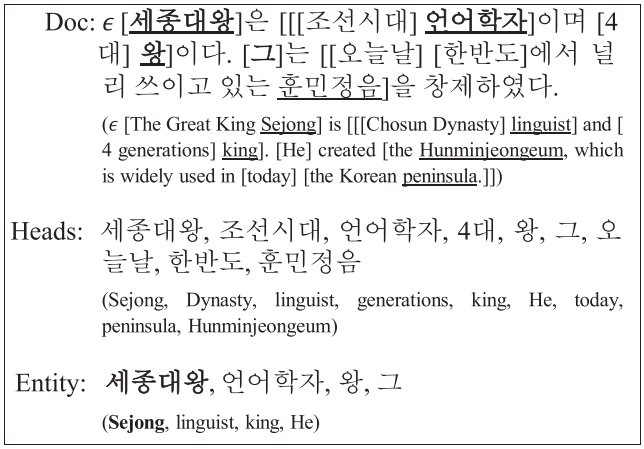

FIGURE 1 Example of a Korean coreference resolution. Mentions are shown in brackets, and ϵ is a dummy symbol in the document (Doc). In mentions containing the modifiers, head is underlined (all heads are nouns). (**Sejong**, **linguist**, **king**, **He**) are coreference resolution words

Here, we propose a simple and effective neural coreference resolution suitable for Korean. (1) In Korean, the head always appears at the end of a mention; therefore, nouns given in a sentence represent mentions and candidates. We propose a head-based model that extracts a list of heads from an input sequence and performs coreference resolution. (2) In the proposed model, when the extracted head finds the referred word, the position of the entity containing the head is output, with the output entity being the position of the first antecedent of the entity containing the given head as input. (3) We perform multitask learning to share the hidden state of mention detection and coreference resolution in bidirectional encoder representation from transformer (BERT) [7] based on the places of hierarchical global attention and the recurrent neural network (RNN) [8] layer that understands the global context. (4) We train a BERT model that creates word representations based on contextual information with a bidirectional transformer in a large-sized Korean corpus and perform fine-tuning to the coreference resolution model. (5) Coreference resolution data are primarily generated at the document level, and to train the model robustly, we perform data augmentation by dividing a given document into sentences.

2 | RELATED WORK

Existing traditional coreference resolution studies are divided into rule-based and statistical-based methods, as well as joint rule- and statistical-based methods. In the rule-based method, Stanford's multipass sieve model [1] applies multipass rules using pronouns, entity attributes, and named-entity information. In Park et al. [9], multipass sieves suitable for Korean were defined to perform Korean coreference resolution. The statistical-based mention pair model [2] is a method that makes reference to another reference using machine learning and by creating a pair comprising the current mention and an arbitrary precedent. Park et al. [10] is a joint model that combines rule-based and statistical-based methods and performs coreference resolution using feed-forward neural networks (FFNNs).

In a recent coreference resolution study, a case was described in which although the heads were the same, the meanings were inconsistent depending on the modifier that describes the head. Hence, Lee et al. [11] defined mentions as span representations and proposed a mention-ranking model based on long short-term memory (LSTM) [12] that considers all spans in the document. Because span representations reflect contextual information, the two mentions are of different entities but are interpreted as similar or related words. This phenomenon results in local consistency errors that yield erroneous coreference resolutions. Therefore, a higher-order method was developed [13] to perform an attention mechanism for mentions and antecedents using a high-order function; this was used in conjunction with a model that weighs antecedents and is related to the current mention. The end-to-end model [11,13] showed superior performance in English coreference resolution; however, the complexity of $O(n^4)$ derives from considering all spans and span pairs in the document. The method proposed by Zhang et al. [14] is based on that of Lee et al. [11] and replaces the concatenated attention score with the biaffine attention score to calculate the conference score. Additionally, it performed the multitask learning process that calculates the loss for the mention score. Moreover, RNN is a type of neural network that can model sequence data. RNNs experience vanishing-gradient problems when the input sequence is long, which makes training difficult. To address this problem and improve performance, models using a gate have been studied. First, LSTM [12] determines the presence or absence of information transmission for each gate using input, output, and forget gates. Gated recurrent units (GRUs) [5] are models that use update and reset gates, thereby reducing computational complexity using only two gates (one less than LSTM). A simple recurrent unit (SRU) [15] is a model that uses three gates (input, forget, and reset), each of which comprises an FFNN with recurrence removed in order to significantly reduce the computation time required and increase the abstraction of the information when stacking layers using a highway network [16] for the output layer. Because SRUs

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

use FFNN-based gates, they exhibit the fastest speeds among RNN types.

BERT has demonstrated excellent performance in various natural language-processing tasks and represents a model trained using a large-sized corpus. To allow applications to other natural language-processing tasks, fine-tuning is performed after adding the output layer to the pretrained BERT model. A BERT-based on a bidirectional transformer [7] trains the language model by verifying the global contextual information at all layers of the network. The model uses masked language modeling (LM) and the next-sentence prediction method for training. Masked LM is a method of masking an arbitrary word in a given sentence based on the contextual information of all networks to allow prediction of the corresponding masked word. A BERT model trained using masked LM creates word representations based on contextual information, and the representation of each word is expressed differently according to the given context. BERT solves the problem of an unknown word by applying byte pair encoding (BPE) [17] to train the language model.

The main contributions of our work are as follows:

- To solve coreference resolution for Korean, which is a head-final language, we propose a simple and effective head-based pointing method accompanied by data augmentation using sentence separation.
- The proposed approach performs multitask learning to share the hidden state of mention detection and coreference resolution using pointer network and biaffine attention.
- We train a BERT model in a large-sized Korean corpus and perform fine-tuning to the coreference resolution model.
- We report a state-of-the-art Korean coreference resolution performance.

3 | TASK FORMULATION

The training data comprised a document input sequence, $X = \{\epsilon, x_1, x_2, ..., x_N\}$, feature sequence for the input sentence, $F = \{f_0, f_1, f_2, ..., f_N\}$ (where f_0 is padding because of the dummy symbol), head-input sequence, $Y_i = \{yi_1, yi_2, ..., yi_M\}$, mention-detection result, $Y_m = \{y_1^m, y_2^m, ..., y_M^m\}$, and coreference resolution result, $Y_c = \{y_1^c, y_2^c, ..., y_M^c\}$. We insert ϵ into the document input sequence to resolve the case where coreference resolution does not occur. In the case of Korean coreference resolution, the nouns of the input sequence are extracted as the head of the mentions, given that the head appears at the end of the noun phrase according to the head-final language. We define the position of the extracted nouns as Y_i, and Y_m, as the result of mention detection is a list of mention-start boundaries and defines the start boundary of the mention corresponding to the given noun in the input sequence. The coreference resolution result, Y_c, is the position in a given input sequence. Because we derive the output result from the attention weight calculation between the input document sequence (length: N) and the input head sequence (length: M), the time complexity for the proposed coreference resolution model is $O(NM)$.

3.1 | Data preprocessing for Korean

Creation of a dictionary in the unit of a word would result in countless vocabularies because Korean is an agglutinative language; therefore, the token of the morpheme unit is mainly used to address this issue. However, out-of-vocabulary (OOV) issues also occur in morpheme units; therefore, we applied BPE to address the OOV problem.

Table 1 shows that input sentences are processed in the input format of the model in a two-step process. Given an input text, morphological analysis is performed using the ETRI language analyzer, and BPE is applied to the analyzed text by morphology. For this method, we use the POS tag together with the results of morphological analysis to specify the POS information of each morpheme. Because Korean is an agglutinative language, a word is divided into morphemes, such as a noun and Josa, which is similar to the postposition. We can explicitly use the meaning of each word according to the morpheme level. To input the text into the model, we apply BPE to the morphemes. Applying BPE, '"세종대왕/NNP' (Sejongdaewang/NNP) and '언어ㅎ자/NNG' (eon-eohagja/NNG) were divided into '"세종' (Sejong), '대왕/NNP_' (daewang/NNP_) and '언' (eon), '어' (eo), '학자/NNG_' (ja/NNG_) according to BPE dictionary matching.

The proposed model is a pointer network-based model comprising an encoder and a decoder. Table 2 shows the training data structure for the input and output sequences of the proposed model. Because the encoder of our model is based on BERT, the input sequence of the encoder uses the result of applying BPE as shown in Table 1, and the [CLS] token is added at the beginning of the sequence, and the [SEP] token is added at the end of the sequence. Table 2A shows an example of 'BPE token [index]' for the input sequence given to the encoder. The decoder is divided into an input sequence and an output sequence. We use nouns extracted from the input document as the input sequence to the decoder.

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 Example of the proposed model's input format

Input sentences	
Korean	세종대왕은 조선시대 언어학자이며 4 대 왕이다. 그는 오늘날 한반도에서 널 리 쓰이고 있는 훈민정음을 창제하였다.
English	The Great King Sejong is Chosun Dynasty linguist and 4 generations king. He created the Hunmminjeongeum, which is widely used in today the Korean peninsula.
pronunciation	sejongdaewang-eun joseonsidae eon-eohagjaimyeo 4 dae wang-ida. geuneun oneulnal hanbando-eseo neol li sseu-igo issneun hunminjeong-eum-eul changjehayeossda.
Step 1: morphological analysis with POS tagging	
Korean	세종대왕/NNP 은/JX 조선/NNP 시대/NNG 언어학자/NNG 이/VCP 며/EC 4/SN 대/NNB 왕/NNG 이/VCP 다/EF ./SF 그/NP 는/JX 오늘날/NNG 한반도/NNP 에서/JKB 널리/MAG 쓰이/VV 고/EC 있/VV 는/ETM 훈민정음/NNP 을/JKO 창제/NNG 하/XSV 았/EP 다/EF ./SF
pronunciation	sejongdaewang/NNP eun/JX joseon/NNP sidae/NNG eon-eohagja/NNG i/VCP myeo/EC 4/SN dae/NNB wang/NNG i/VCP da/EF./SF geu/NP neun/JX oneulnal/NNG hanbando/NNP eseo/JKB neol-li/MAG sseu-i/VV go/EC iss/VV neun/ETM hunminjeong-eum/NNP eul/JKO changje/NNG ha/XSV ass/EP da/EF./SF
Step 2: applying BPE	
Korean	세종 대왕/NNP_ 은/JX_ 조선/NNP_ 시대/NNG_ 언 어 학자/NNG_ 이/VCP_ 며/EC_ 4/SN_ 대/NNB_ 왕/NNG_ 이/VCP_ 다/EF_ ./SF_ 그/NP_ 는/JX_ 오 늘 날/NNG_ 한반도/NNP_ 에서/JKB_ 널리/MAG_ 쓰이/VV_ 고/EC_ 있/VV_ 는/ETM_ 훈 민 정 음/NNP_ 을/JKO_ 창 제/NNG_ 하/XSV_ 았/EP_ 다/EF_ ./SF_
pronunciation	sejong daewang/NNP_ eun/JX_ joseon/NNP_ sidae/NNG_ eon eo hagja/NNG_ i/VCP_ myeo/EC_ 4/SN_ dae/NNB_ wang/NNG_ i/VCP_ da/EF_./SF_ geu/NP_ neun/JX_ o neul nal/NNG_ hanbando/NNP_ eseo/JKB_ neol-li/MAG_ sseu-i/VV_ go/EC_ iss/VV_ neun/ETM_ hun min jeong eum/NNP_ eul/JKO_ chang je/NNG_ ha/XSV+ ass/EP_ da/EF_./SF_

Note: The Korean BERT we use uses a large-capacity corpus that is morphologically analyzed to reflect the characteristics of Korean. Table 1 shows an example changing the given raw text to the input format of the BERT. In the first row, we show the raw text of the Korean input sentences and represent the English translation and pronunciation for the Korean sentences. Step 1 is a result of analyzing morpheme and POS tag for the input sentence, and we use concatenating each morpheme and POS tag. In Step 2, we apply BPE to the morphologically analyzed results to match the input format of the BERT.
Abbreviations: BERT, bidirectional encoder representation from transformer; BPE, byte pair encoding.

Because BPE is applied to the input sequence, we define the first subtoken of the morpheme corresponding to the noun as decoder input. For example, the morpheme '세종대왕/NNP' (sejong daewang) is divided into '"세종' (sejong) and '대왕/NNP_' (daewang) by applying BPE, and the index of each token is 1 and 2. We use the first subtoken '"세종' (sejong) as input to the decoder. Internally, the decoder extracts hidden states of the position corresponding to the noun from hidden states of the input sequence and uses the extracted hidden state as input. Table 2B shows the output of the decoder as a position where each input of the decoder corresponds to the encoder sequence. For example, when we assume that '-' (eon) corresponding to index 6 in the encoder represents input to the decoder, we define '"종' (sejong), which is a word with a resolved coreference with '-' (eon), as an output target. The model uses index 1 in the encoder as the output of the decoder. We define words that cannot be coreferent, such as '조선/NNP_' (joseon), and '시대/NNG_' (sidae), to an output target as index 0, which is used as the token [CLS].

3.2 | Features

In this study, we used five features for Korean coreference resolution: word boundary (word), morpheme boundary (morp), dependency parsing (dep), named-entity recognition (NER), and head distance (dst). We create feature representations by including all of the features in one dictionary. Additionally, we use an ETRI language analyzer for dependency parsing and NER. The description of each feature is as follows:

- Word boundary: This is used to study the boundary feature of the coreference resolution in word units. The starting token of the word is divided into *B*, and the following token is divided into *I* tags.

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 Input and output examples

Encoder: BERT input sequence
Korean:
[CLS][0] 세종[1] 대왕/NNP_[2] 은/JX_[3] 조선/NNP_[4] 시대/NNG_[5] 언[6] 어[7] 학자/NNG_[8] 이/VCP_[9] 며/EC_[10] 4/SN_[11] 대/NNB_[12] 왕/NNG_[13] 이/VCP_[14] 다/EF_[15] ./SF_[16] 그/NP_[17] 는/JX_[18] 오[19] 늘[20] 날/NNG_[21] 한반도/NNP_[22] 에서/JKB_[23] 널리/MAG_[24] 쓰이/VV_[25] 고/EC_[26] 있/VV_[27] 는/ETM_[28] 훈[29] 민[30] 정[31] 음/NNP_[32] 을/JKO_[33] 창[34] 제/NNG_[35] 하/XSV_[36] 았/EP_[37] 다/EF_[38] ./SF_[39] [SEP][40]
Pronunciation:
[CLS][0] sejong[1] daewang[2] eun[3] joseon[4] sidae[5] eon[6] eo[7] hagja[8] i[9] myeo[10] 4[11] dae[12] wang[13] i[14] da[15].[16] geu[17] neun[18] o[19] neul[20] nal[21] hanbando[22] eseo[23] neol-li[24] sseu-i[25] go[26] iss[27] neun[28] hun[29] min[30] jeong[31] eum[32] eul[33] chang[34] je[35] ha[36] yeoss[37] da[38].[39] [SEP][40]

(A) Example of BERT input sequence for encoder

Decoder			
input noun	**input position**	**output token**	**output position**
세종 sejong	1	세종 sejong	1
조선/NNP_ joseon	4	[CLS]	0
시대/NNG_ sidae	5	[CLS]	0
언 eon	6	세종 sejong	1
4/SN_ 4	11	[CLS]	0
왕/NNG_ wang	13	세종 sejong	1
그/NP_ geu	17	세종 sejong	1
오 o	19	[CLS]	0
한반도/NNP_ hanbando	22	[CLS]	0
훈 hun	29	[CLS]	0

(B) Example of input and output for decoder

Note: Each token and its index for the input document of encoder (A) are shown. In decoder (B), we show the BPE subtoken of the morpheme corresponding to the noun and the location information for each token in the encoder. The input noun column shows nouns with BPE applied. The input position column describes the position of each token in the encoder, we use this index as the input of the decoder. The output token and position columns show the output result corresponding to each input of the decoder.

Abbreviations: BERT, bidirectional encoder representation from transformer; BPE, byte pair encoding.

- Morpheme boundary: This is used to reflect the morpheme-boundary characteristics of the morpheme-analysis results. Morp-*B* is the beginning token, and morp-*I* is the inside token of the morpheme.
- Dependency parsing: We use the dependency parsing label as a feature to reflect the structural and semantic information of the sentences.
- NER: This uses type information for each entity appearing in the document as a feature.
- Head distance: This represents distance information between extracted candidate nouns, with the distance measured from the immediately preceding noun using the following buckets [1, 2, 3, 4, 5–7, 8–15, 16–31, 32–63, 64+] [18].

4 | PROPOSED MODEL

To solve coreference resolution, the proposed model employs a pretrained BERT, as shown in Figure 2. The input sequence, which is analyzed for a morpheme, is tokenized for BPE, and [CLS] and [SEP] are added as the

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

first and last tokens, respectively, according to the input format of the pretrained BERT data. The output of Figure 2 represents the start boundary position of the mention and a position of the antecedent referenced with the mention. If a mention boundary or a reference position needs to be solved in the input sequence, the corresponding position is noted and subsequently output. Otherwise, the position of the first token [CLS] of the input sequence is output. To identify the mention-start boundary and the coreference resolution target, we define the first token among the tokens, which are divided by applying BPE to the morphemes, as a pointing reference (for example, "'"종 (Sejong)," "o/NNG_ (king)" in Figure 2).

4.1 | Encoder with BERT

After morpheme analysis, the input sequence tokenized by BPE is transformed into the input representation, E_i, for BERT by adding token embedding, segment embedding, and position embedding. Subsequently, token representation, T_i, is obtained from pretrained BERT data through the transformer and concatenated with the feature representation for coreference resolution. Based on this, a bidirectional RNN (BiRNN) [8,19] is applied to create the hidden state, $\mathbf{r}_i$, as shown in (1).

$$\begin{aligned} f_i &= [f_i^{\text{word}}; f_i^{\text{morp}}; f_i^{\text{dep}}; f_i^{\text{NER}}; f_i^{\text{dst}}] \\ x_i &= [T_i; f_i] \\ \mathbf{r}_i &= \text{BiRNN}(\mathbf{r}_{i-1}, x_i). \end{aligned} \tag{1}$$

4.2 | Head encoder with a self-matching layer

We extract the hidden state corresponding to the position of the head-input sequence, yi_j, from the encoded hidden state, $\mathbf{r}_i$, in order to create the hidden state for the head. To encode the contextual information between heads, the BiRNN is applied for the head hidden state (that is, $\mathbf{r}_j^{yi} = \text{BiRNN}(\mathbf{r}_{j-1}^{yi}, \mathbf{r}_{yi_j})$). Subsequently, the hidden state, $\mathbf{r}_j^{yi}$, is concatenated with the context vector, $\mathbf{c}_j$, of the self-matching layer to allow deeper abstraction and encoded as a gated attention-based recurrent network [20]. In (2), the gated attention-based recurrent network generates a vector by determining the element-wise sum between the input of the network and the sigmoid-applied nonlinear gate layer. This allows for a more significant vector value to be increased.

$$\begin{aligned} \mathbf{r}_j^h &= \text{BiRNN}\left(\mathbf{r}_{j-1}^h, \left[\mathbf{r}_j^{yi}; \mathbf{c}_j\right]^*\right), \\ \mathbf{g}_j &= \text{sigmoid}\left(\mathbf{W}_g\left[\mathbf{r}_j^{yi}; \mathbf{c}_j\right]\right), \\ \left[\mathbf{r}_j^{yi}; \mathbf{c}_j\right]^* &= \mathbf{g}_j \& \left[\mathbf{r}_j^{yi}; \mathbf{c}_j\right]. \end{aligned} \tag{2}$$

The self-matching layer enables the application of an attention mechanism to a given input, such as itself. The attention weight is adjusted by increasing the score between words of similar meaning in the input sequence. The formulas for the self-matching layer are as follows:

$$\begin{aligned} \mathbf{c}_j &= \sum_{k=1}^{m} \alpha_{j,k} \mathbf{r}_j^h, \\ \alpha_{j,k} &= \exp(s_{j,k}) / \sum_{t=1}^{m} \exp(s_{j,t}), \\ s_{j,t} &= \mathbf{r}_j^h \mathbf{W}_h \mathbf{r}_t^h. \end{aligned} \tag{3}$$

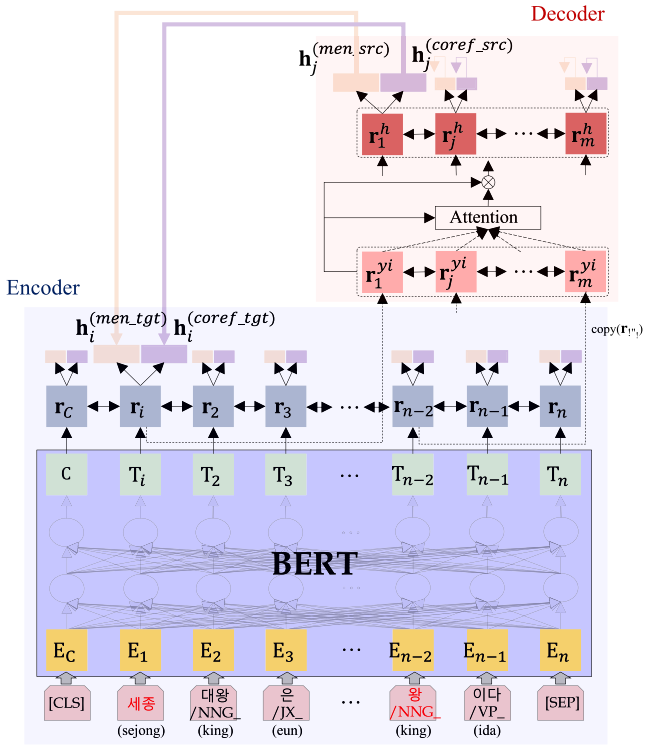

FIGURE 2 Korean coreference resolution model using bidirectional encoder representation from transformer (BERT)-based deep attention. The proposed model uses an simple recurrent unit (SRU) to obtain hidden states, which are encoded by concatenating BERT embeddings for the input sequence using five-feature embeddings in the encoder. To generate more abstracted hidden states, the decoder performs SRU modeling and self-attention to determine the hidden states of nouns extracted from encoder hidden states. Decoder outputs are computed to predict the (1) mention-start boundary and (2) coreference resolution with hidden states of the encoder

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.3 | Deep Biaffine and Bilinear Attention for the Output

According to (4), we apply a nonlinear operation to each jth hidden state of the head-input sequence (similar to [21]) in order to output the solved coreference position. The activation function used here is "elu" [22].

$$\begin{aligned}\mathbf{h}_i^{(men\_src)} &= \text{elu}(\text{FFNN}^{(men\_src)}(\mathbf{r}_i))\\ \mathbf{h}_i^{(men\_tgt)} &= \text{elu}(\text{FFNN}^{(men\_tgt)}(\mathbf{r}_i))\\ \mathbf{h}_i^{(coref\_src)} &= \text{elu}(\text{FFNN}^{(coref\_src)}(\mathbf{r}_i))\\ \mathbf{h}_i^{(coref\_tgt)} &= \text{elu}(\text{FFNN}^{(coref\_tgt)}(\mathbf{r}_i)).\end{aligned} \tag{4}$$

We use biaffine and bilinear attentions to perform the attention mechanism and output the mention-start boundary and coreference resolution results according to the following formulas:

$$\begin{aligned}s_{j,i}^{(men)} &= \mathbf{h}_i^{T(men\_tgt)}\mathbf{U}\mathbf{h}_j^{(men\_src)} + \mathbf{w}^T\mathbf{h}_j^{(men\_src)}\\ s_{j,i}^{(coref)} &= \mathbf{h}_i^{T(coref\_tgt)}\mathbf{U}\mathbf{h}_j^{(coref\_src)} + \mathbf{w}^T\mathbf{h}_j^{(coref\_src)}.\end{aligned} \tag{5}$$

In this equation, the biaffine model calculates the attention score using matrix multiplication for the *src* and *tgt* hidden states with the bias term. To output the mention boundary, we calculate the biaffine for $\mathbf{h}_i^{(men\_tgt)}$ in $\mathbf{h}_j^{(men\_src)}$, as shown in (5). We calculate biaffine for $\mathbf{h}_i^{(coref\_tgt)}$ in $\mathbf{h}_j^{(coref\_src)}$ in order to output the coreference resolution result. Removing the bias term from biaffine attention is a bilinear attention operation. The computed attention score, $s_{j,i}$, applies softmax to output the probability of the mention-start boundary or the antecedent position of the coreference resolution. When performing multitask learning, the training for mention detection and coreference resolution uses a cross-entropy loss, as shown in (6).

$$\mathcal{L}_{\text{loss}} = -\sum_i^n y_i^m \log \hat{y}_i^m - \sum_i^n y_i^c \log \hat{y}_i^c, \tag{6}$$

where y_i^m is the cross-entropy loss for start boundary output of a mention and y_i^c is the cross-entropy loss for output of a coreference resolution.

5 | EXPERIMENTS

5.1 | BERT: Korean pretraining

BERT comprises a bidirectional transformer encoder with several layers. In this study, we used a pretrained model with BERT-based hyperparameters, including $L = 12$, $H = 768$, and $A = 12$; a hidden layer and attention layer dropout of 0.1; gelu as an activation function; and cased for Latin alphabets. To pretrain the BERT model, we used Google BERT [7] (refer to the open source[1]) and news and Wikipedia data (a total of 23.5 GB) collected from the internet. After performing morpheme analysis on all input words, we concatenated the results as "morpheme/POS tag", and tokenization was performed with subwords using BPE. The dictionary comprised 30 349 BPE tokens. BERT is an encoder of the transformer and performs masked LM and next-sentence prediction to understand the relationship between two sentences. We applied a method of arbitrarily modifying 15% of all BPE tokens to mask tokens according to [7]. Among the modified tokens, 80% were changed to [MASK] tokens, 10% were changed to random tokens from the vocabulary, and the remaining 10% were left unchanged. To perform next-sentence prediction, we concatenated two sentences in order and trained IsNext and NotNext using the sequence. For this training, the second sentence represented an actual next sentence with a 50% probability and a random sentence from the corpus with a 50% probability. The BERT model used the ETRI language analyzer, which is available in AIOpen[2], for morpheme analysis. The ETRI language analyzer is a tool for natural language processing of Korean (such as NLTK and SpaCy) and performs morphological analysis, POS tagging, NER, dependency parsing, and semantic role labeling (SRL). We performed morphological analysis using this language analyzer to divide the input document into morpheme units, as well as obtain POS tagging, NER, and dependency parsing results, for use as features.

5.2 | Dataset and evaluation

We used the ETRI quiz-domain coreference resolution dataset used in Park et al. [23] to evaluate coreference resolution. The ETRI dataset comprises 2819 training sets, 645 development sets, and 571 test sets. We used three standard metrics (MUC [24], B^3 [25], and CEAF_{ϕ_4} [26]) and reported the F1 for each metric and the average precision, recall, and F1 as the final CoNLL score using the official CoNLL-2012 evaluation scripts [27]. Because Korean is a head-final language, we measured coreference resolution using only the head of the mentions.

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

5.3 | Implementation details

We set the dropout as 0.1 and optimized the number of hidden layer dimensions and feature dimensions of the RNN to 800 and 1600, respectively. We optimized the RNN layer. The learning rate was set to 5e-5, and the linear method was used for the learning rate schedule. The training algorithm used Adam [28], with the Adam weight decay set to 1e-02. The batch size was set to 6, and the epoch number was set to 70. The maximum length of the input sequence was limited to 430, because the most extended input sequence length in the test set was 424.

5.4 | Overall performance

Table 3 shows a comparison of the proposed model with state-of-the-art methods for Korean coreference resolution. To compare the performance on the test set, we present the performances of the rule-based model [10], the end-to-end neural model [11], and the higher-order model [13] together. For the proposed model, the performance of removing mention-detection attention (Our model−M.D.), data augmentation (+split n sentences), and ensemble revealed the best results according to all F1 metrics (the single model achieved 64.7% F1). Use of a multilingual pretrained model of Google BERT returned an F1 of 63.6% with a 1.1% difference. For the single model, removal of mention-detection attention and the corresponding loss decreased the F1 by 0.8%, and multitask learning facilitated solving the coreference resolution. To augment the data, the sentence-separation method divides one sentence into two, three, four, and five sentences, each of which is accumulated (refer chapter 5.5). The proposed model improved the overall performance of data augmentation in the single model. For increasing the data divided into two sentences, the proposed model showed an F1 of 67.7%, which exceeded the percentages returned by data augmentation models. Additionally, the ensemble model performing data augmentation using a single model splits a single sentence into two to five sentences and demonstrated the best performance (F1 = 68.5%).

5.5 | Sentence-separation method for data augmentation

The size of the training set used in this study was small. To address this limitation, we augmented the data by applying a sentence-separation method using a sliding window. This method splits the given input document into window sizes, n ($2 \leq i \leq n$), and strides (that is, 1 in this study). For the present study, we split the document by applying a window size ranging from two to n and returned the split documents to the original document. Table 3 shows an example of data augmentation, and Table 3 shows the original data as a single document comprising four sentences. We set the window size to three to divide the document repeatedly, with the augmented dataset comprising both original and split data.

TABLE 3 Experimental results using a test set of Korean data from ETRI wiseQA

	MUC			B^3			$CEAF_{\phi_4}$			CoNLL
Model	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Avg. F1**
Rule-based [10]	52.0	45.1	48.3	51.6	45.4	48.3	53.8	51.9	52.9	49.8
e2e-coref [11]	66.9	55.2	60.5	64.5	53.1	58.2	66.1	53.9	59.4	59.4
c2f-coref [13] (optimized)	68.3	56.4	61.8	59.0	53.4	59.0	66.4	54.4	59.8	60.2
Our model − M.D.	64.9	67.3	66.0	60.7	**66.0**	63.3	60.4	64.7	62.5	63.9
Our model (single)	66.4	65.5	65.9	63.3	64.6	64.0	63.8	64.6	64.2	64.7
+split 2 sentences	70.7	**67.4**	69.0	68.5	64.8	66.6	72.1	63.3	67.4	67.7
+split 3 sentences	70.7	64.7	67.5	68.5	62.1	65.1	**72.4**	61.1	66.3	66.3
+split 4 sentences	68.2	65.3	66.7	66.3	64.0	65.1	65.4	64.7	65.1	65.7
+split 5 sentences	69.7	65.3	67.4	66.9	63.8	65.3	67.7	63.9	65.8	66.2
Our model (ensemble)	**72.6**	66.9	**69.6**	**70.3**	65.5	**67.8**	70.4	**66.1**	**68.2**	**68.5**

Note: The final column (CoNLL Avg. F1) is the main evaluation metric averaged by the F1 of MUC, B^3, and $CEAF_{\phi_4}$. Based on the Korean head-final, the coreference resolution score is calculated based on the head of the mentions. Bold text indicates the highest performance in the Avg. F1 score.

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

6 | ANALYSIS

6.1 | Comparison of the attention-score function

We then compared the performance of each attention-score function. Table 4 shows that the biaffine attention-score function of 0.43% was better than the bilinear attention-score function in terms of average F1. The bias term is beneficial for coreference resolution.

6.2 | Optimizing RNN types

The RNN types are LSTM, GRU, and SRU. In Dozat and Manning [21], a deep biaffine was performed based on LSTM. Table 5 shows that the RNN type was suitable for coreference resolution; therefore, we optimized the number of layers for that type. The experimental results indicated that the SRU performed better than LSTM or the GRU. Because the SRU uses a highway network [16], a skip connection is used to allow the gradient to directly propagate to the previous layer, resulting in minimal information loss, even if the stack deepens. The optimal number of SRU layers was two, which resulted in an F2 of 68.44%.

TABLE 4 Comparison of the optimized hyperparameter of the attention mechanism using a one-layer bi-LSTM on the development set

Attention	MUC	B^3	$CEAF_{\phi_4}$	Avg. F1
Biaffine	70.34	67.42	65.77	**67.84**
Bilinear	69.64	67.13	65.70	67.49

Note: Bold text indicates the highest performance in the Avg. F1 score.
Abbreviation: LSTM, long short-term memory.

TABLE 5 Optimizing the RNN type and stack number using the development set

RNN type	Layer	Avg. F1
LSTM	1	67.84
LSTM	2	67.91
GRU	1	67.71
GRU	2	67.12
SRU	1	67.92
SRU	2	**68.44**

Note: Bold text indicates the highest performance in the Avg. F1 score.
Abbreviations: GRU, gated recurrent unit; LSTM, long short-term memory; RNN, recurrent neural network; SRU, simple recurrent unit.

6.3 | Optimizing the dimension size of RNN hidden layers

The RNN representation differs depending on the number of dimensions, which affects coreference resolution; therefore, we optimized the number of dimensions based on the attention-score function and the RNN type (Tables 4 and 5, respectively). The number of dimensions used for optimization was 100, 200, 400, 800, and 1600. We found that the optimal number of dimensions for the RNN hidden layer was 800 F1 = 68.44%) (Table 6).

6.4 | Feature ablation study

To understand the effect of features on the proposed model, we performed feature ablation using the development set. As shown in Table 7, removing the distance feature showed a smaller decrease (0.06%) in coreference resolution relative to that of the single model. We subsequently removed the features one at a time and found that removing the NER resulted in a 0.7% reduction resolution in the single model, and removal of the dependency and word boundary features resulted in further decreases of 0.04%. Removal of all features showed a result 0.96% lower than that in the single model. These results indicated that removing the NER feature significantly affected the coreference resolution relative to other features.

TABLE 6 Optimizing the dimension size of RNN hidden layers using the development set

Dimension size of the RNN hidden layer	Avg. F1
100	67.49
200	68.08
400	67.97
800	**68.44**
1600	67.36

Note: Bold text indicates the highest performance in the Avg. F1 score.
Abbreviation: RNN, recurrent neural network.

TABLE 7 Evaluation of the effects of feature ablation using the development set

Feature	Avg. F1	Δ
Our model (single)	68.44	-
−head distance	68.38	−0.06
−NER	67.74	−0.70
−dependency	67.70	−0.74
−word boundary	67.66	−0.78
−morp boundary	67.48	−0.96

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

7 | CONCLUSION

In summary, we described a head-based end-to-end model using head-final Korean attributes and used to perform neural coreference resolution for Korean. We applied BERT, which was pretrained using a large-sized Korean corpus, to the proposed model to solve the unknown word problem and use context-based word representations. The results returned an average F1 of 64.7% for the test set and a performance difference of 12.1% relative to the proposed model (single). These findings suggested the utility of using a fine-tuned BERT for Korean coreference resolution. One of the contributions of this study is demonstration of the efficacy of using a pretrained 23.5-GB Korean corpus and fine-tuned BERT for coreference resolution for Korean. To address the limitation of small amount of data, we added sentences via data augmentation. The proposed model achieved state-of-the-art performance on Korean coreference resolution and exhibited a time complexity of $O(n^2)$ based on performance of an attention operation for a given input sequence and head noun sequence. In future work, we will include a method to reduce the number of output classes by using a target sequence as a head-input sequence for coreference resolution and apply a mask that allows only antecedents to be calculated when an attention score is computed.

ACKNOWLEDGMENTS

This research was supported by the This work was supported by Institute for Information & Communications Technology Promotion (IITP) grants funded by the Korean government (MSIT) (2013-0-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services).

AUTHOR CONTRIBUTIONS

Cheoneum Park and Changki Lee designed the model. Cheoneum Park carried out the experiments and wrote the manuscript. All authors provided critical feedback and helped shape the research, analysis and manuscript.

ORCID

Cheoneum Park https://orcid.org/0000-0001-5386-0483

REFERENCES

1. H. Lee et al., *Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task*, in Proc. Conf. Comput. Natural Lang. Learn.: Shared Task (Portland, OR, USA), June 2011, pp. 28–34.
2. V. Ng and C. Cardie, *Improving machine learning approaches to coreference resolution*, in Proc. Annu. Meet. Assoc. Comput. Linguistics (Philadelphia, PA, USA), July 2002, pp. 104–111.
3. W. M. Soon, H. T. Ng, and D. C. Y. Lim, *A machine learning approach to coreference resolution of noun phrases*, Comput. Linguistics **27** (2001), no. 4, 521–544.
4. N. Kwon, M. Polinsky, and R. Kluender, *Subject preference in Korean*, in Proc. West Coast Conf. Form. Linguistics (Somerville, MA, USA), Sept. 2006, pp. 1–14.
5. D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, arXiv preprint, CoRR, 2014, arXiv: 1409.0473.
6. A. Vaswani et al., *Attention is all you need*, in Proc. Conf. Neural Inf. Process. Syst. (Long Beach, CA, USA), Dec. 2017, pp. 5998–6008.
7. J. Devlin et al., *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint, CoRR, 2018, arXiv: 1810.04805.
8. T. Mikolov et al., *Recurrent neural network based language model*, in Proc. Annu. Conf. Int. Speech Commun. Assoc. (Chiba, Japan), Sept. 2010, pp. 1045–1048.
9. C. Park, K.-H. Choi, and C. Lee, *Korean coreference resolution using the multi-pass sieve*, J. KIISE **41** (2014), no. 11, 992–1005.
10. C. Park et al., *Korean coreference resolution with guided mention pair model using deep learning*, ETRI J. **38** (2016), no. 6, 1207–1217.
11. K. Lee et al., *End-to-end neural coreference resolution*, CoRR, 2017, arXiv: 1707.07045.
12. S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Comput. **9** (1997), no. 8, 1735–1780.
13. K. Lee, L. He, and L. Zettlemoyer, *Higher-order coreference resolution with coarse-to-fine inference*, arXiv preprint, 2018, arXiv: 1804.05392.
14. R. Zhang et al., *Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering*, arXiv preprint, CoRR, 2018, arXiv: 1805.04893.
15. T. Lei, Y. Zhang, and Y. Artzi, *Training RNNS as fast as CNNS*, arXiv preprint, CoRR, 2017, arXiv: 1709.02755.
16. R. K. Srivastava, K. Greff, and J. Schmidhuber, *Highway networks*, arXiv preprint, CoRR, 2015, arXiv: 1505.00387.
17. R. Sennrich, B. Haddow, and A. Birch, *Neural machine translation of rare words with subword units*, arXiv preprint, CoRR, 2015, arXiv: 1508.07909.
18. K. Clark and C. D. Manning, *Improving coreference resolution by learning entity-level distributed representations*, arXiv preprint, CoRR, 2016, arXiv: 1606.01323.
19. M. Schuster and K. K. Paliwal, *Bidirectional recurrent neural networks*, IEEE Trans. Signal Process. **45** (1997), no. 11, 2673–2681.
20. W. Wang et al., *Gated self-matching networks for reading comprehension and question answering*, in Proc. Annu. Meet. Assoc. Comput. Linguistics (Vancouver, Canada), July 2017, pp. 189–198.
21. T. Dozat and C. D. Manning, *Deep biaffine attention for neural dependency parsing*, arXiv preprint, CoRR, 2016, arXiv: 1611.01734.
22. D.-A. Clevert, T. Unterthiner, and S. Hochreiter, *Fast and accurate deep network learning by exponential linear units (ELUs)*, arXiv preprint, CoRR, 2015, arXiv: 1511.07289.

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

23. C. Park et al., *Contextualized embedding- and character embedding-based pointer network for Korean coreference resolution*, Annu. Conf. Hum. Lang. Technol. **2018** (2018), 239–242.
24. M. Vilain et al., *A model-theoretic coreference scoring scheme*, in Proc. Conf. Message Underst. (Columbia, Maryland), Nov. 1995, pp. 45–52.
25. A. Bagga and B. Baldwin, *Algorithms for scoring coreference chains*, in Proc. Int. Conf. Lang. Resour. Eval. (Granada, Spain), May 1998, pp. 563–566.
26. X. Luo, *On coreference resolution performance metrics*, in Proc. Conf. Hum. Lang. Technol. Empirical Methods Nat. Lang. Process. (Vancouver, Canada), Oct. 2005, pp. 25–32.
27. S. Pradhan et al., *Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes*, in Proc. Joint Conf. EMNLP and CoNLL: Shared Task (Jeju Island, South Korea), July 2012, pp. 1–40.
28. D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv preprint, CoRR, 2014, arXiv: 1412.6980.

AUTHOR BIOGRAPHIES

Cheoneum Park received his BS, MS, and PhD degrees in Computer Science from Kangwon National University, Chuncheon, Republic of Korea, in 2014, 2016, and 2020, respectively. He is now a senior research engineer at AIRS Company of Hyundai Motor Group, Seoul, Republic of Korea. His research interests include natural language processing, natural language understanding, question answering, and deep learning.

Joonho Lim received BS and MS degrees in Computer Science from Korea University, Seoul, South Korea, in 2002 and 2005, respectively. He received his PhD in Computer Engineering from Chungnam National University, Daejeon, South Korea, in 2016. He is now a principal researcher at Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include deep learning language model, question answering, and natural language understanding.

Jihee Ryu received a BS degree in Computer Science & Engineering from Chungnam National University, Daejeon, South Korea, in 2006. He received an MS degree in Computer Science from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2010. He is now a researcher at Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include deep learning language model, question answering, and natural language understanding.

Hyunki Kim Received BS and MS degrees in Computer Science from Chunbuk National University, Chonju, South Korea, in 1994 and 1996, respectively. He received a PhD in Computer Engineering from the University of Florida, USA, in 2005. He is now a principal researcher at Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include natural language processing, question answering, and deep learning.

Changki Lee received a BS degree in Computer Science from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 1999. He received an MS degree and PhD in Computer Engineering from POSTECH, Pohang, Republic of Korea, in 2001 and 2004, respectively. From 2004 to 2012, he was a researcher with the Electronics and Technology Research Institute (ETRI), Daejeon, Republic of Korea. Since 2012, he has been a professor of Computer Science at Kangwon National University, Chuncheon, Republic of Korea. His research interests include natural language processing, machine learning, and deep learning.

22337326, 2021, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2020-0282 by Electronics and Telecommunications, Wiley Online Library on [03/09/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License