K-EPIC: Entity-Perceived Context Representation in Korean Relation Extraction

Yuna Hur 1,†, Suhyune Son 1,†, Midan Shim 2,3,†, Jungwoo Lim 1,† and Heuiseok Lim 1,*

1 Department of Computer Science and Engineering, Korea University, 145, Anam-ro, Seongbuk-gu, Seoul 02841, Korea; yj72722@korea.ac.kr (Y.H.); ssh5131@korea.ac.kr (S.S.); wjddn803@korea.ac.kr (J.L.)
2 Human-Inspired AI Research, Korea University, 145, Anam-ro, Seongbuk-gu, Seoul 02841, Korea; hihello0426@gmail.com
3 Department of Software Convergence, Kyung Hee University, Yongin 17104, Korea
* Correspondence: limhseok@korea.ac.kr
† These authors contributed equally to this work.

Abstract: Relation Extraction (RE) aims to predict the correct relation between two entities from the given sentence. To obtain the proper relation in Relation Extraction (RE), it is significant to comprehend the precise meaning of the two entities as well as the context of the sentence. In contrast to the RE research in English, Korean-based RE studies focusing on the entities and preserving Korean linguistic properties rarely exist. Therefore, we propose K-EPIC (Entity-Perceived Context representation in Korean) to ensure enhanced capability for understanding the meaning of entities along with considering linguistic characteristics in Korean. We present the experimental results on the BERT-Ko-RE and KLUE-RE datasets with four different types of K-EPIC methods, utilizing entity position tokens. To compare the ability of understanding entities and context of Korean pre-trained language models, we analyze HanBERT, KLUE-BERT, KoBERT, KorBERT, KoELECTRA, and multilingual-BERT (mBERT). The experimental results demonstrate that the F1 score increases significantly with our K-EPIC and that the performance of the language models trained with the Korean corpus outperforms the baseline.

Keywords: information extraction; relation extraction; Korean pre-trained language model; deep learning

1. Introduction

The importance of research on automatic information extraction is recently increasing [1] with the advent of massive unstructured documents. Information Extraction (IE), which provides the basic research for extracting structured information from unstructured resources, is considered promising research in the field of natural language processing (NLP). Among principal research in IE [2], Relation Extraction (RE) aims to predict the relation between two entities in a single sentence. RE is a significant task especially in the field of Knowledge Base Population (KBP) since it extracts structured triples. Additionally, it is used for advanced research, including Question and Answering (QA) systems, Summarization, Dialogue Systems, and Information Retrieval (IR) [3].

To obtain the final prediction form as (entity1, relation, entity2) in RE tasks, it is important to determine the precise meaning of two entities along with the context of the sentence [2]. For example, as shown in Figure 1, it is much easier to predict the relation “org: place_of_headquarters” if a person already knows the meaning of the subject “한국방송공사에서 (Korea Broadcasting Corporation)” marked in orange and “대한민국의 (South Korea)”, the object entity marked in purple. In a recent study, pre-trained language models trained on the large corpus to capture contextual information, such as BERT [4], show considerable performance in various NLP tasks including RE. In the ways of utilizing pre-trained language model in RE tasks, one represents entities by aligning sentence and
entities separated with [SEP] token [5], or another replaces entities with corresponding Named Entity Recognition (NER) tags [6]. The former implicitly represents the sentences without any entity-specific tokens, and the latter merely replaces two entities with specific tokens, such as <entity 1>, resulting in losing their meaning. To overcome these problems, Soares et al. [7] utilizes the explicit representation of entities and predicts the relation on an English RE dataset [8,9].

Soares et al. [7] utilizes the explicit representation of entities and predicts the relation on an English RE dataset [8,9].

Figure 1. Example of Korean RE. The translation of the sentence in Korean is “KBS 1 Radio is a radio channel operated and broadcasted by the Korea Broadcasting Corporation (Entity 1) in South Korea (Entity 2)”.

With the previous RE research in English, BERT-Ko-RE [10] and KLUE-RE [11], which are recently published data on Korean RE, elevate the level of research in Korean. In addition, Nam et al. [10] demonstrates the performance of multilingual-BERT (mBERT) [4] trained with the corpus in 104 languages. However, a limited number of RE studies considering two linguistic properties of Korean which are different from English exist. One is that the role of the word in most Korean sentences is decided when the postposition is fully combined [12]. For instance, by combining ‘총리 (root word; the prime minister)’ and ‘는 (postposition)’, the word ‘총리는’ finally acts as the subject. Therefore, it is important to extract the comprehensive meaning of the word which is a combined form of root word and postposition. The other property is the free word order in Korean sentences. Contrary to English, the order of words is not a substantial obstacle to comprehending the meaning of the sentence, since the role of phrase or word is decided according to the types of a postposition [13]. Owing to the aforementioned characteristics, language models that have not been trained with Korean corpus exhibit limited performance on Korean RE.

In this paper, we propose entity-perceived context representation and Korean language models. To apply our K-EPIC method to the Korean pre-trained language models, the entities of the dataset are marked with entity position tokens. Since we aim to predict relation by capturing the meaning of entities considering the linguistic properties especially in Korean, we demonstrate our experiments on the BERT-Ko-RE-dataset [10] and KLUE-RE dataset [11]. We conduct experiments with Korean language models, including HanBERT (https://github.com/monologg/HanBert-Transformers (accessed on 1 March 2021)), KLUE-BERT [11], KoBERT (https://github.com/SKTBrain/KoBERT (accessed on 1 March 2021)), KorBERT (https://aiopen.etri.re.kr/service_dataset.php (accessed on 1 March 2021)), and KoELECTRA [14] to enhance their capability of understanding context and entity representations simultaneously. By utilizing language models trained on Korean corpus, our model exhibits better performance than those of previous methods while preserving linguistic characteristics of Korean in Relation Extraction (RE) tasks. We also analyze the results of mBERT to compare the capability of comprehending Korean as well in an empirical way. To the best of our knowledge, this is the first work that proposes an entity aware method in RE achieving improved performance on five Korean language models.

The contributions of our work are summarized as follows:

- We propose K-EPIC, four different types of representing entities in Korean Relation Extraction.
- We apply our K-EPIC method to the five Korean Pre-trained Language Models (PLMs) and analyze each result empirically.
- Significant improvements are shown in the experiments when we apply our K-EPIC method to the Korean PLMs.
2. Related Works

2.1. Relation Extraction

RE (Relation Extraction) is the task of obtaining triple elements \((entity1, relation, entity2)\) by predicting the relation when the single sentence and two entities are given. In English, feature-based methods [15–17] that do not use pre-trained language models classify the relation using Support Vector Machine (SVM) algorithm with the information of Part-of-Speech (POS) tagging or syntactic parse trees. Lin et al. [18] predict relations employing the word embeddings and position embeddings passed from a Convolutional Neural Network (CNN). Soares et al. [7], the first pre-trained model-based approach, utilizes special tokens to manage entity representations to predict relation from the sentence. Similarly, Zhou and Chen [6] exploits special tokens to mark entities and shows the impacts on Named Entity Recognition (NER) feature on relation representation. Moreover, this research described two ways of utilizing special tokens.

In previous Korean RE studies, Kim and Lee [19] extracts the entities from the sentence using Part of Speech (POS) tagging, Named Entity Recognition, and word embeddings. Then, using the result of dependency parsing from the constructed data, Bayesian probability is applied to predict the final relation. Kim et al. [20] directly constructs the Korean text about the history and predicts the relation using the Long Short-Term Memory (LSTM) model.

However, few studies have been done in the Korean language model-based method on the BERT-Ko-RE dataset as well as KLUE-RE. Therefore, we propose the K-EPIC method that enables us to analyze the impact of entity position tokens in diverse ways on RE tasks in Korean.

2.2. Pre-Trained Models

Pre-trained models trained with the large corpus show considerable performance recently in diverse Natural Language Processing (NLP) tasks. As a leading trend of pre-trained language models, BERT [4] uses Transformer encoder [21] with 3.3 billion tokens exploiting random masking strategy. Various models based on BERT have been state of the art in many downstream tasks, and BERT is still used to compare performance for evaluating as a baseline. ELECTRA [22] is proposed with a more efficient pre-training task which consists of a generator and discriminator network that are similar structures with Generative Adversarial Network (GAN). ELECTRA is much faster and efficient since it learns diverse features from the input tokens.

In Table 1, we denote the detailed information of Korean pre-trained language models including multilingual-BERT. HanBERT uses its own tokenizer (Moran) with 54,000 vocabulary words from 70 GB of Korean general documents and patent documents. KLUE-BERT is the pre-trained language model that is open to the public along with the KLUE dataset. KLUE-BERT is trained with morpheme-based subword tokenizer, 32,000 vocabulary words from [23] and 63 GB sentences including Modu corpus (https://corpus.korean.go.kr/ (accessed on 1 March 2021)), CC-100 (http://data.statmt.org/cc-100/ (accessed on 1 March 2021)), NamuWiki (https://namu.wiki/ (accessed on 1 March 2021)), newspaper, and other web sources. KoBERT trained on 5 M sentences from Korean Wikipedia (https://ko.wikipedia.org/ (accessed on 1 March 2021)) and uses SentencePiece [24] for tokenizer with 8002 vocabulary words. Additionally, KorBERT is trained with 23 GB of text from Korean news and an encyclopedia. KorBERT is also providing two different versions of tokenizers, which are wordpiece-based tokenizers (30,797 vocabulary words) and morpheme-based tokenizers (30,349 vocabulary words). KoELECTRA is proposed to improve previous Korean pre-trained language models with 32,200 vocabulary words. KoELECTRA uses a WordPiece tokenizer and is pre-trained with 34 GB of data from Korean Wikipedia, NamuWiki, newspaper, and Modu corpus.
2.3. Tokenizer in Korean Pre-Trained Language Models

Unlike English tokenization as in BERT, a tokenization strategy reflecting the linguistic characteristics of Korean is significant. In Korean, the word generally contains more information than the word in English since the word in Korean is in the combined form of the root word and postposition [25]. Postposition includes 은(eun), 는(neun), 이(i), 가(ga), 를(leul), and ي(uyi). Owing to the fact that the word in a Korean sentence represents the meaning when the postposition is combined, the order of words has relatively low impact on also understanding the sentence. Due to the above characteristics, it is rather important to comprehensively understand the word while capturing the morphological relationship between the root word and postposition in Korean tokenization. Likewise, HanBERT, KLUE-BERT, and KorBERT properly tokenize the sentence as depicted in Table 2. Another way of understanding the word comprehensively without a morpheme-based tokenizer is training the language models in Korean only. For instance, mBERT and KoELECTRA tokenize sentences into subword units differently even though they are using the same WordPiece tokenizer strategy. Due to its difference in pre-training data, mBERT just divides 총리(' into 총리') and '##' failing to distinguish the root word from postposition. On the other hand, KoELECTRA splits of word 총리(' into the root word 총리' and postposition '##') correctly. While a mBERT tokenizer was trained with data in 104 languages, including Korean, the KoELECTRA tokenizer is trained on Korean Wikipedia, Namu Wiki, and other Korean documents focusing on the Korean language. As a result, the KoELECTRA tokenizer reflects the characteristics of Korean rather than that of mBERT. Likewise, HanBERT, KLUE-BERT, KoBERT, and KorBERT that trained with a Korean corpus separate the word 총리(' into the root word 총리' and postposition '##'). Therefore, it is important to apply a morpheme-based tokenizer or utilize the Korean pre-trained language model to process the unique properties of Korean. To this end, we apply a morpheme-based tokenizer to our K-EPIC method with Korean pre-trained language models.

Table 2. Comparison on the results of tokenization from each model. #, ~, and _ denote the boundary symbol of each tokenizer.

<table>
<thead>
<tr>
<th>Model</th>
<th>Model Size</th>
<th>Types of Pre-Trained Data</th>
<th>Pre-Trained Data Size</th>
<th>Tokenizers</th>
<th>_vocab_size</th>
</tr>
</thead>
<tbody>
<tr>
<td>mBERT</td>
<td>178 M</td>
<td>Wikipedia pages in 104 languages</td>
<td>2.5 B words</td>
<td>WordPiece</td>
<td>119,547</td>
</tr>
<tr>
<td>HanBERT</td>
<td>128 M</td>
<td>General Document, Patent Document</td>
<td>70 GB (11.3 B morphemes)</td>
<td>Moran</td>
<td>54,000</td>
</tr>
<tr>
<td>KLUE-BERT</td>
<td>111 M</td>
<td>Modu corpus, CC-100-Kor, Namu Wiki, News Crawl, Petition</td>
<td>63 GB</td>
<td>Morpheme-based subword tokenizer</td>
<td>32,000</td>
</tr>
<tr>
<td>KoBERT</td>
<td>92 M</td>
<td>Korean Wikipedia</td>
<td>5 M sentences, 54 M words</td>
<td>SentencePiece</td>
<td>8002</td>
</tr>
<tr>
<td>KoBERT</td>
<td>110 M</td>
<td>Newspaper, Encyclopedia</td>
<td>23 GB (4.7 B morphemes)</td>
<td>Morpheme/WordPiece</td>
<td>30,349 (morphemes), 30,797 (wordpiece)</td>
</tr>
<tr>
<td>KoELECTRA</td>
<td>112 M</td>
<td>Korean Wikipedia, Namu Wiki, Newspaper, Messages, Web, etc.</td>
<td>34 GB</td>
<td>WordPiece</td>
<td>35,000</td>
</tr>
</tbody>
</table>
3. Proposed Method

We propose an K-EPIC method for Korean RE tasks as depicted in Figure 2. We preprocess datasets to indicate the position of entities and analyze the four different methods of utilizing the entities.

![Figure 2. Overview of our K-EPIC process utilizing RE.](image)

3.1. Task Definition

Consider the dataset \( D = \{ (S_0,e_1,0,e_2,0,r), \ldots, (S_N,e_1,N,e_2,N,r) \} \), where \( N \) is the number of examples in dataset. Each sentence \( S_n \) is labeled with two entities \( e_1 \) and \( e_2 \), and relation \( r \in \{ r_0, \ldots, r_R \} \), where \( R \) is the total number of the relations. The sentence can be denoted as \( S_n = [t_1^n, t_2^n, \ldots, t_m^n] \). \( t_i^n \) is the \( i \)-th token of \( S_n \), and \( m \) denotes the number of tokens from the tokenizer. Moreover, \( t_1^n \) and \( t_m^n \) are [CLS] token and [SEP] token, respectively. The entity position tokens, \([e_1\text{SP}],[e_1\text{EP}],[e_2\text{SP}],[e_2\text{EP}]\), are added at the start and end of both entities, i.e., \( e_1 \) and \( e_2 \). Therefore, the Sentence Input (SI) can be represented by Equation (1). \( e_1 \) indicates the number of tokens in \( e_1 \); it can also be applied to \( e_2 \) in the same manner. The position of \( e_1 \) and \( e_2 \) can be switched according to the relation type.

\[
SI = \left[ [\text{CLS}] + t_1, \ldots, t_i + [e_1\text{SP}] + e_1 + [e_1\text{EP}] + t_{i+e_1+2}, \ldots, t_j + [e_2\text{SP}] + e_2 + [e_2\text{EP}] + t_{j+e_2+2}, \ldots, t_m-1 + [\text{SEP}] \right]
\]  

(1)

3.2. Relation Extraction with K-EPIC

We extend the methods of providing additional entity information in four ways, inspired by the work of Soares et al. [7]. Our four types of K-EPIC methods are illustrated in Figure 3. We put \([e_1\text{SP}]\) and \([e_1\text{EP}]\) at the start and end positions of the subject entity. Similarly, we place \([e_2\text{SP}]\) and \([e_2\text{EP}]\) at the start and end positions of the object entity, respectively. Because \( e_1 \) does not always precede \( e_2 \) in all sentences, our K-EPIC identifies the position of each entity and inserts the entity position tokens before and after the entity. Except for nonK-EPIC, we use this final preprocessed sentence, SI, as an input. Detailed information regarding each K-EPIC is explained below.

**nonK-EPIC** nonK-EPIC representation merely utilizes the [CLS] token of the input sentence in the final layer of language models. Note that nonK-EPIC inputs do not include any entity position tokens. As the hidden states barely include information about entity position, the models are reluctant to find the precise position of entities and their meanings explicitly. nonK-EPIC uses \( h_V \) as the final hidden representation, which is identical to the hidden state vector of the [CLS] token.

**K-EPIC\textsubscript{V}** Whereas nonK-EPIC is not created from SI, the vanilla method of our K-EPIC, K-EPIC\textsubscript{V} processes [CLS] of the SI. Similar to nonK-EPIC, K-EPIC\textsubscript{V} uses \( h_V \) as the final hidden representation. We can compare the effect of entity position tokens between nonK-EPIC and K-EPIC\textsubscript{V}.

**K-EPIC\textsubscript{S}** K-EPIC\textsubscript{S} exploits \( h_S \), which represents the concatenation of the start position token of each entity, \([e_1\text{SP}]\) and \([e_2\text{SP}]\). The concatenated vector is then fed into the linear projection layer to obtain the final relation. The representation enables the models to perceive entity information.

**K-EPIC\textsubscript{E}** In contrast to K-EPIC\textsubscript{S}, K-EPIC\textsubscript{E} uses only end position tokens, i.e., \([e_1\text{EP}]\) and \([e_2\text{EP}]\), in the same manner as K-EPIC\textsubscript{S}. In K-EPIC\textsubscript{E}, the final hidden representation, \( h_E \), is the concatenation of these two end position tokens.

**K-EPIC\textsubscript{SE}** We also suggest the combination representation of K-EPIC\textsubscript{S} and K-EPIC\textsubscript{E} that utilizes all position tokens of the entities in SI, i.e., \([e_1\text{SP}],[e_1\text{EP}],[e_2\text{SP}],[e_2\text{EP}]\). K-EPIC\textsubscript{SE} uses \( h_{SE} \), which is also the concatenated result of all entity position tokens. We
analyzed the performance differences between these methods based on the number of
entity position tokens.

In summary, the input of nonK-EPIC consists of only original sentence with [CLS]
and [SEP] tokens. SI, which includes entity position tokens, is applied to K-EPICV, K-
EPICs, K-EPICe, and K-EPICse. The input embedding, segment embedding, and position
embedding of the SI are used to create the representation. The final representation, \( h \), is
then fed into the linear layer and projected on the size of total relations. Consequently,
the softmax result denotes the final probability for predicting the relation, as shown in
Equation (2):

\[
Pr(r_{SI}) = \text{softmax}(W^T \cdot h + b)
\] (2)

where \( W \) and \( b \) are learnable parameters.

**Figure 3.** Four different types of final representations. In the example sentence, “Korea Broadcasting Corporation” is the
subject entity and “South Korea” is the object entity. K-EPICV only utilizes the [CLS] token, whereas K-EPICs and K-EPICe
utilize the start position token and end position token, respectively. K-EPICse exploits all entity position tokens, which are
the start position tokens and end position tokens of both entities.

### 4. Experiments

#### 4.1. Datasets

We demonstrate our experiments with our K-EPIC method on two existing RE datasets.
The statistics of each dataset can be found in Table 3.

<table>
<thead>
<tr>
<th></th>
<th>BERT-Ko-RE</th>
<th>KLUE-RE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Test</td>
</tr>
<tr>
<td>Number of Sentences</td>
<td>20,603</td>
<td>1838</td>
</tr>
<tr>
<td>Number of Relations</td>
<td>49</td>
<td>29</td>
</tr>
</tbody>
</table>
4.1.1. BERT-Ko-RE Dataset

BERT-Ko-RE [10] is a Korean RE dataset annotated in two different ways. One is a crowdsourcing dataset created in compliance with the Gated Instruction (GI) protocol [26] and the other is made in a Distant-Supervision (DS) approach [27]. The crowdsourcing dataset only includes the data tagged with high agreement from the workers, and it guarantees high-quality notations, unlike DS data that contain a high rate of noise. The dataset contains pairs of a sentence and its corresponding relation label. Each relation is defined in an ontology schema with short English labels, such as country, knownFor, and part, etc. that are grounding daily domain. As described in Table 3, the train and test sets have 20,603 and 1838 pairs, respectively, with 49 relations.

4.1.2. KLUE-RE Dataset

The KLUE-RE dataset [11] is also a hand-crafted Korean RE dataset based on the Korean knowledge base (https://aihub.or.kr/aidata/84 (accessed on 1 March 2021)), Wikipedia, and NamuWiki. As the official test set is not yet disclosed, we use the development set as the test set. A single data example includes an original sentence, relation label, data source, and information indicating a subject or object. Entity labels contain common relations related to daily life, such as member_of, place_of_birth, and parents. The train and test sets have 32,470 and 7765 pairs, respectively, with 30 relations, as presented in Table 3.

4.2. Experimental Setting

We perform our experiments on the BERT-Ko-RE [10] and KLUE-RE datasets [11]. Using the proposed K-EPIC, we compare the capabilities of different pre-trained language models for RE tasks with different entity position tokens to demonstrate their enhanced performance. To ensure a high-quality dataset, we only use the hand-crafted datasets for the experiments. Our model trains on a single RTX 8000 GPU; the learning rate and batch size are set to be 2e-5 and 2, with 12 attention heads and 10 epochs. Moreover, we use the same parameters for all six models and take cross-entropy as the loss function.

4.3. Results and Analysis

We utilize two evaluation metrics, micro-F1 and weighted-F1. Most existing RE studies [3,28] use micro-F1 as an evaluation metric, since the datasets contain class-specific data imbalance. Furthermore, we also employ the weighted-F1 which considers the ratio of data by class. In this section, we introduce three different analyses in the respect of types of K-EPIC methods, language models, and tokenizers.

4.3.1. Comparison on the K-EPIC Method

The performance of the proposed K-EPIC on BERT-Ko-RE [10] and KLUE-RE dataset [11] are depicted in Tables 4 and 5, respectively. First, the proposed K-EPIC_V method demonstrates a significant increase of 35.97%p for weighted-F1, on average, in comparison to nonK-EPIC. The difference illustrates that the entity position tokens enhance the RE performance. We also demonstrate that K-EPIC_S achieves the best performances in mBERT, KLUE-BERT, KoBERT, and KorBERT, among the other methods in all metrics. In the case of HanBERT, K-EPIC_S obtains the highest micro-F1, and K-EPIC_SE presents the highest weighted-F1, with 73.72 % and 78.62 %, respectively. KoELECTRA, which has a different model architecture, achieves the highest scores in K-EPIC_V.

Table 5 demonstrates the results on the KLUE-RE dataset. Similar to previous experimental results, the performance on our K-EPIC_V increases by 32.44%p for weighted-F1, on average, including all language models, in comparison to nonK-EPIC. Similarly, HanBERT, KLUE-BERT, and KoBERT show the best performance when they utilized K-EPIC_S. The rest of the language models perform better when they exploit K-EPIC_SE.

From our comparative analysis, we find that the inclusion of the start position token has a positive impact on the final performance. We assume that the closeness to the
entity is related to the result. Since Korean postposition is usually located next to the root word, the start position token has more proximity of the entity in the input of the sentence [29]. As a result, K-EPIC$S$ mostly achieves the best performance among five Korean pre-trained language models, whereas the performance utilizing K-EPIC$SE$ may have been obstructed by the end position token obtaining dissimilar representation to the entity. These results show that our K-EPIC$S$ method makes efficient predictions with the entity position information.

Table 4. Experimental results on the BERT-Ko-RE dataset on six language models within the K-EPIC method and average \%p is calculated by subtracting nonK-EPIC from the average of weighted-F1 score of all of the language models (the best performing part is additionally marked in bold).

<table>
<thead>
<tr>
<th>Data</th>
<th>BERT-Ko-RE dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>mBERT  HanBERT  KLUE-BERT  KoBERT  KorBERT  KoELECTRA</td>
</tr>
<tr>
<td>Metric (F1)</td>
<td>Micro</td>
</tr>
<tr>
<td>nonK-EPIC</td>
<td>39.39</td>
</tr>
<tr>
<td>K-EPIC$V$</td>
<td>70.40</td>
</tr>
<tr>
<td>K-EPIC$S$</td>
<td>73.56</td>
</tr>
<tr>
<td>K-EPIC$E$</td>
<td>69.97</td>
</tr>
<tr>
<td>K-EPIC$SE$</td>
<td>70.29</td>
</tr>
</tbody>
</table>

Table 5. Experimental Results on the KLUE-RE dataset on six language models within the K-EPIC method and average \%p is calculated by subtracting nonK-EPIC from the average of weighted-F1 score of all of the language models (the best performing part is additionally marked in bold).

<table>
<thead>
<tr>
<th>Data</th>
<th>KLUE-RE Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>mBERT  HanBERT  KLUE-BERT  KoBERT  KorBERT  KoELECTRA</td>
</tr>
<tr>
<td>Metric (F1)</td>
<td>Micro</td>
</tr>
<tr>
<td>nonK-EPIC</td>
<td>20.91</td>
</tr>
<tr>
<td>K-EPIC$V$</td>
<td>53.43</td>
</tr>
<tr>
<td>K-EPIC$S$</td>
<td>54.27</td>
</tr>
<tr>
<td>K-EPIC$E$</td>
<td>55.18</td>
</tr>
<tr>
<td>K-EPIC$SE$</td>
<td>55.78</td>
</tr>
</tbody>
</table>

4.3.2. Comparison on the Korean Language Model

To effectively compare the language models with K-EPIC$S$ and K-EPIC$SE$ which include start position token, Figure 4 illustrates the performance of each dataset. As shown in the above figures, it is evident that KLUE-BERT outperforms other language models on both datasets. Moreover, despite its promising performance, HanBERT has slight differences from the performance of KLUE-BERT, implying that the volume of pre-trained data has a primary impact on downstream tasks.

For the BERT-Ko-RE dataset, even though HanBERT trained the largest corpus than any other language model, the slightly low performance in RE tasks can be attributed to the type of data it trained. Since the patent documents usually include legal terminology for the protection of technologies [30], we assume that it has little impact on the RE tasks, which usually contain daily expressions.

For the KLUE-RE dataset, KLUE-BERT exhibits the best performance owing to the relevancy of its pre-trained data and the RE dataset. In addition, Korean-based language models such as KLUE-BERT exhibit comparable performances despite having less trained datasets than mBERT. Similar to the results of the BERT-Ko-RE dataset, HanBERT presents the second-highest score for micro-F1 because of its volume of pre-training data.

4.3.3. Comparison on Korean Tokenizers

Due to the fact that the Korean language is significantly different from English, the impact of tokenizers should be examined as well. KorBERT, which provides two different
tokenizers on the same model architecture, we compare them under our K-EPIC method. One is the KorBERT model with word-level tokenizers and the other is that with the morpheme-level tokenizers. We conduct the experiments on the both BERT-Ko-RE and KLUE-RE datasets with the K-EPIC method. As indicated in Table 6, the model with morpheme-level tokenizers achieves a significantly higher score than that with word-level tokenizers in both datasets. The results can be attributed to the linguistic property of the Korean language, which is an agglutinative language where words may contain different postpositions according to the role of the word [31]. Along with these results, K-EPIC_{SE} shows the best performance in comparison to other K-EPIC methods and nonK-EPIC in most cases.

Figure 4. (a) Performance on each language model on the BERT-Ko-RE dataset. The green and pink bars indicate the performance of K-EPIC_{S} and K-EPIC_{SE}, respectively; (b) performance on each language model on the KLUE-RE dataset. The green and pink bars indicate the performance of K-EPIC_{S} and K-EPIC_{SE}, respectively.

Table 6. Experimental results between different tokenizers on KorBERT. Underlined text denotes the highest score of the KorBERT with a morpheme-based tokenizer and bold text indicates the highest score among the methods of utilizing K-EPIC.

<table>
<thead>
<tr>
<th>Data</th>
<th>BERT-Ko-RE Dataset</th>
<th>KLUE-RE Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>KorBERT w/Wordpiece</td>
<td>KorBERT w/Morpheme</td>
</tr>
<tr>
<td>Metric (F1)</td>
<td>Micro</td>
<td>Weighted</td>
</tr>
<tr>
<td>nonK-EPIC</td>
<td>32.26</td>
<td>33.89</td>
</tr>
<tr>
<td>K-EPIC_{V}</td>
<td>58.32</td>
<td>61.86</td>
</tr>
<tr>
<td>K-EPIC_{S}</td>
<td>60.45</td>
<td>64.43</td>
</tr>
<tr>
<td>K-EPIC_{E}</td>
<td>57.02</td>
<td>59.60</td>
</tr>
<tr>
<td>K-EPIC_{SE}</td>
<td>61.59</td>
<td>65.69</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, we propose K-EPIC, four different methods of representing entities in Korean Relation Extraction. To consider a Korean linguistic property as well as entity representation, we employ most Korean pre-trained language models and achieve enhanced performance rather than mBERT. The experimental results demonstrate that, by using entity position tokens, the capability of understanding the entities of pre-trained language models significantly improved. We evaluate the methods in each language model and find that K-EPIC_{S} shows the best performance in most language models. Finally, we present the experimental results on the comparison of different tokenizers indicating that morpheme-level tokenization impacts the performance effectively. In conclusion, our work may suggest...
that the performance of a model in Korean RE highly depends on the understanding level of the entities in sentences while preserving Korean linguistic properties.

**Author Contributions:** Conceptualization, Y.H.; methodology, Y.H.; software, S.S.; validation, S.S.; formal analysis, Y.H. and J.L.; investigation, M.S.; writing—original draft preparation/review and editing, Y.H., S.S., M.S. and J.L.; visualization, M.S. and J.L.; funding acquisition/project administration/supervision, H.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1A6A1A03045425). Additionally, it was also supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Creative Consilience program (IITP-2021-2020-0-01819) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. These data can be found here: “https://github.com/machinereading/bert-ko-re (accessed on 1 March 2021)” and “https://github.com/KLUE-benchmark/KLUE (accessed on 1 March 2021)”.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**


