



Text Data Augmentation for the Korean Language

Dang Thanh Vu ¹, Gwanghyun Yu ¹, Chilwoo Lee ² and Jinyoung Kim ^{1,*}

¹ Department of ICT Convergence System Engineering, Chonnam National University, 77, Yongbong-ro, Buk-gu, Gwangju 500757, Korea; dtvu1707@gmail.com (D.T.V.); sayney1004@gmail.com (G.Y.)

² Department of Computer Engineering, Chonnam National University, 77, Yongbong-ro, Buk-gu, Gwangju 500757, Korea; leecw@chonnam.ac.kr

* Correspondence: beyondi@jnu.ac.kr

Abstract: Data augmentation (DA) is a universal technique to reduce overfitting and improve the robustness of machine learning models by increasing the quantity and variety of the training dataset. Although data augmentation is essential in vision tasks, it is rarely applied to text datasets since it is less straightforward. Some studies have concerned text data augmentation, but most of them are for the majority languages, such as English or French. There have been only a few studies on data augmentation for minority languages, e.g., Korean. This study fills the gap by demonstrating several common data augmentation methods and Korean corpora with pre-trained language models. In short, we evaluate the performance of two text data augmentation approaches, known as text transformation and back translation. We compare these augmentations among Korean corpora on four downstream tasks: semantic textual similarity (STS), natural language inference (NLI), question duplication verification (QDV), and sentiment classification (STC). Compared to cases without augmentation, the performance gains when applying text data augmentation are 2.24%, 2.19%, 0.66%, and 0.08% on the STS, NLI, QDV, and STC tasks, respectively.

Keywords: data augmentation; language modeling; Korean language processing



Citation: Vu, D.T.; Yu, G.; Lee, C.; Kim, J. Text Data Augmentation for the Korean Language. *Appl. Sci.* **2022**, *12*, 3425. <https://doi.org/10.3390/app12073425>

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 25 February 2022

Accepted: 24 March 2022

Published: 28 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Machine learning models often depend on training data size and quality. Training a machine learning model on underrated datasets will make it lose generality and lead to poor performance. However, a good-quality dataset is prohibitively expensive to collect and annotate. In this context, automatic data augmentation is regularly used to obtain more robust models, particularly when only a deficient dataset is available [1]. Data augmentation in computer vision is basically performed based on human inductive bias, e.g., “the class of object should be unchanged under affine transformations, such as rotation or translation”. However, data augmentation for text data has rarely been studied directly, and it turns out to be even more problematic for languages other than English.

It has been widely observed that natural language processing (NLP) relies on large-scale manually annotated training sets to achieve state-of-the-art performance. However, such datasets are scarce or non-existent in specific NLP tasks for uncommon languages (e.g., Korean). Recent works have suggested several techniques for data augmentation in NLP. A common approach is to generate new data by translating sentences into French and back to English [2,3]. Besides that, it was also claimed that text could be effectively augmented by simple paraphrasing [4]. Some authors have recently promoted the further development of using pre-trained language models (PLMs) to generate text conditional on instructions and labels [5–7].

Data augmentation in computer vision can be properly performed by way of transformations like resizing, random cropping, or color shifting. Unfortunately, in the case of natural languages, it often poses a dilemma because ensuring semantic compatibility of contexts and labels is not trivial [6]. There has been less previous evidence of generalized

rules for language data augmentation techniques in NLP. Existing data augmentation methods for text, developed with manual lexical rules, often lose generality. For example, generating new sentences by replacing their words with relevant ones can only produce limited patterns from the original texts because very few words have exactly or nearly the same meanings [4]. In addition, due to differences in task requirements and unique language grammar, text data augmentation is a domain-specific problem. For example, in reading comprehension, the goal of data augmentation is to generate questions accurately with a given reference passage; conversely, creating questions in commonsense reasoning normally requires creativity [8]. However, there are growing appeals for improved data efficiency in generative settings via pre-trained language models [7,9–11]. These techniques are not often preferred in practice because the implementation costs are high considering the performance gains.

Our research aims to investigate language data augmentation under the usage of domain-specific settings. The contribution of this work is twofold:

- We experiment on the Korean language with two tasks: semantic textual similarity (STS) and natural language inference (NLI). Both are supervised tasks that rely heavily on gold labeling. Also, we show the effect of data augmentation on duplication verification (QDV) and sentiment classification (STC).
- We experiment with two data augmentation practices: Easy Data Augmentation (EDA) [4], which combines simple transformations on text data, and back translation (BT) [12], an unsupervised approach that generates new sentences by utilizing another language.

The rest of this article is structured as follows: Section 2 reviews the fundamentals of data augmentation and pre-trained models and corpora of Korean. Section 3 then briefly introduces our formulation of data augmentation and the methods we used in this study (i.e., EDA and BT). Section 4 demonstrates our experiment settings and reports our experimental results, while Section 5 concludes this study.

2. Related Works

2.1. Related Studies on Data Augmentation

Machine learning models are often overfitted and lose generality due to small training datasets. Data augmentation deals with data scarcity by generating more training samples [1]. Nearly all modern computer vision advances benefit from data augmentation to allow the model to view more diverse examples and learn invariant to non-semantic changes in input. On the other hand, augmentation is infrequently used in Natural Language Processing. Nevertheless, several methods were reported in previous studies to address this issue.

Wei et al. presented a set of simple universal text transformation techniques for NLP called Easy Data Augmentation (EDA) [4], including synonym replacement, random insertion, random swap, and random deletion. Along with EDA, back translation [12,13] commonly serves as a baseline in various research due to its simplicity [5,6]. However, these techniques suffer from several weaknesses: back translation assumes word co-existence between the source and target languages, which is not often the case in practice, while EDA often produces false sentence grammar or changes the original meaning.

The use of pre-trained language models for sentence generation has been successfully established, as described in Refs [5–7]. Kobayashi et al. used a fill-in-the-blank strategy to augment data by replacing words in a sentence with blanks and inferring the words using a seq2seq model given conditions on that sentence's label [5]. Inspired by Kobayashi's approach, Wu et al. reported better results using a pre-trained BERT instead of seq2seq models [6]. Kumar et al. tested on a variety of pre-trained language models, including an autoencoder BERT [13], a seq2seq2BART [14], and an auto-regressive GPT-2, to generate new words with guaranteed condition consistency of the sentence context and its labels [7]. These approaches can retain the context of synthesized sentences but depend on powerful pre-trained models, which are often not available for domain-specific tasks.

Recent research has suggested that data augmentation on a language can be approached by few-shot learning [9,11] and knowledge distillation [10]. Timo et al. introduced Dataset from DINO Instructions (DINO) [11] and GENPET [9]; both methods are capable of automatically generating sentence pair datasets of arbitrary size by providing a PLM with textual instruction. In order not to break the label compatibility of generated data, an alternative to contextual enhancement is to re-label it. Reimers et al. proposed Augmented SBERT (AugSBERT) [10], which uses a BERT cross-encoder to label randomly selected input pairs and then adds them to the training set for the SBERT bi-encoder [15]; the result was a significant performance increase.

2.2. Korean Corpora and Their Pre-Trained Language Models

The training of large neural networks with the goal of language modeling has made significant improvements to NLP tasks. Typically, Bidirectional Encoder Representations from Transformers (BERT) [13] is a pre-trained deep bidirectional representation with a masked language model and the next sentence prediction objective. BERT is based on Transformer [16], an attention-based backbone that provides structured memory to handle long-term dependencies in a sequential structure. BERT uses cross-encoders to perform full attention on input pairs. On the other hand, the bi-encoder used in SentenceBERT [15] maps each input independently to a dense vector space. While cross-encoders typically achieve higher performance, they are extremely computationally expensive for many tasks. On the other hand, the bi-encoder is computationally efficient, but it often achieves lower performance than the BERT cross-encoder.

Machine translation needs large parallel corpora of paired sentences in the source and target languages. However, bitext is limited, while a larger amount of monolingual data exists [12]. Study [3] presented a single attention-based seq2seq NMT model to translate between multiple languages. Their solution did not require changing the model architecture from the standard NMT system but instead introduced a dummy token at the beginning of the input sentence to identify the required target language. Using a common lexicon between languages, their approach allows multilingual NMT to use a single model with significant performance gains, despite no increase in parameters. Furthermore, extension from the monolingual model to the multilingual model can be accomplished by knowledge distillation [17]. The training is based on the idea that a pair of source and target sentences must be mapped to the same location in vector space. The authors used the monolingual model as a teacher to generate embedded sentences for the source language and then train a new system as a student model on the translated sentences to mimic the teacher model. However, multilingual models demand a huge vocabulary, so they are often a burden on deriving tasks, such as fine-tuning on a particular language.

Korean is often referred to in the research community as a low-resource language, due to the lack of resources and inadequate advertising and curation [18]. Compared to the industrial demand, the interest in Korean natural language processing has not yet received widespread attention from academic research. Korean is an agglutinative language, so there will be more problems in vocabulary expression because of its morphological diversity [19]. Korean tokenizes most words as single characters rather than morpheme-like units, and each token is unlikely to be separated by a “space” character, as in English. Furthermore, Korean consists of more than 10,000 syllabic characters, requiring both a vocabulary and an encoder to efficiently handle a variety of complex word forms. These characteristics limit the ability to derive downstream tasks from a pre-trained model in languages other than Korean itself.

Several Korean language datasets have been reported in the recent literature to address the issue discussed here. Regarding natural language comprehension tasks, for example, Ham et al. [20] introduced KorNLI and KorSTS for natural language inference (NLI) and semantic textual similarity, respectively. These two datasets were constructed using neural machine translation from English and partly by manual translation and re-labeling. Although there are a few datasets that come with scientific reports on them, most Ko-

rean language datasets are published scattered across the internet. To address this issue, Cho et al. systematically described Korean corpora in a practical report [18].

The pre-trained language model for Korean is also incomplete, though a few publications providing a pre-trained model are available—for instance, KoreALBERT [21] and BERT [19]. Aside from pre-trained models with international publications, several pre-trained models are available online for the Korean language, though lacking scientific reports. Table 1 summarizes the models mentioned earlier, considering their pre-trained dataset and performance on downstream tasks.

Table 1. Performance of Korean pre-trained models on various downstream tasks.

Model/Task	SKT-koBERT ¹	SKT-GPT2 ²	KrBERT ³	ENLIPIE-v2 ⁴	KoELECTRA-Base-v3 ⁵	BERT-Base ⁶
Pretrained language model						
Pretrained dataset	25 M sentences	40 GB of text	20 M sentences	174 M sentences	180 M sentences	100 GB of text
Pretrained topic	Wikipedia	Wikipedia	Wikipedia + news	Wikipedia + news	News comments	Reviews + blog
Tokenizer	Sentence Piece	character BPE	Bidirectional WordPiece	WordPiece	WordPiece	WordPiece
Vocab size (word)	8 K	51 K	16 K	32 K	30 K	42 K
Finetuned dataset (downstream task)						
NSCM (Sentiment analysis) ⁷	90.10	93.3	89.84		90.63	90.87
koSTS (Semantic textual similarity) ⁸	79.64	78.4		84.75	85.53	84.31
KorQuad (Question-answering) ⁹	80.27		89.18	91.77	93.45	89.45
KorNER (Named entity recognition) ¹⁰	86.11		64.50		88.11	87.27
KorNLI (Natural language inference) ¹¹	79.00			83.21	82.24	82.32

¹ <https://github.com/SKTBrain/KoBERT>. ² <https://github.com/SKT-AI/KoGPT2> (5 February 2022). ³ https://github.com/snunlp/KR-BERT/tree/master/krbert_pytorch. ⁴ https://github.com/enlipleai/kor_pretrain_LM. ⁵ <https://github.com/Beomi/KcELECTRA>. ⁶ <https://github.com/kiyoungkim1/LMkor> (20 February 2022). ⁷ <https://github.com/e9t/nsmc>. ⁸ <https://github.com/kakaobrain/KorNLUDatasets>. ⁹ <https://korquad.github.io/>. ¹⁰ <https://github.com/toriving/naver-nlp-challenge-2018>. ¹¹ <https://github.com/kakaobrain/KorNLUDatasets> (18 February 2022).

3. Data Augmentation

In general, text data augmentation aims to generate a synthetic set of text data not identical to the original set that helps enhance the capability of a learning model. This study utilizes the EDA approach, a set of simple text transformations operating at the word level. Additionally, we employ back translation as another text data augmentation technique that generates sentence-level synthetic examples.

To formulate, we assume a pre-set $S_{pre} = \{X_{pre}, Y_{pre}\}$, where X_{pre} is a set of training documents $x^i = x_1 x_2 \dots x_n \in X_{pre}$ with maximum length n , and x_i is an element of the sentence, such as word or token, including special tokens (e.g., [SEP], [BOS], [EOS], [PAD]) in the case where x^i consists of multiple sentences. While Y_{pre} contains corresponding labels of x^i , $y^i \in Y_{pre}$, depending on the specific task, y^i can be presented as a sentence (e.g., textual similarity) or an integer label (e.g., text classification). Data augmentation aims to build a supplementary set $S_{aug} = \{X_{aug}, Y_{aug}\}$ of synthetic examples generated from the pre-documents that are not identical to the original ones. Concretely, a sample $(x_{aug}^i, y_{aug}^i) \in X_{aug} \times Y_{aug}$ of the augmented set is sampled from a generator $p_{\theta}(x_{aug}^i, y_{aug}^i | x_{pre}^i, y_{pre}^i)$ that parameterizes from a statistic of the pre-set $\theta = \theta(S_{pre})$ or a set of prior knowledge $\theta = \theta(P_1 \dots P_n)$, where P_i is a premise. A premise P_i is a clue or condition to accept or reject a sample generated from p_{θ} . In this study, we consider the two particular generators below.

3.1. EDA: Easy Data Augmentation

Jason et al. proposed four augmentation operations that are loosely inspired by computer vision. Those techniques include synonym replacement (SR), random insertion (RI), random swap (RS), and random deletion (RD) [4]. The synthesized examples x_{aug} are constructed from $x_{pre} = x_1x_2 \dots x_n$ by combining the following methods:

- **Synonym Replacement:** The premise here is to randomly select words in the sentence x_{pre} that are neither stop words nor special tokens, and replace each of these words with one of its synonyms at random, i.e., $x_{aug} = x_1x_2 \dots syn(x_{k_i}) \dots x_n$, where $k_i \leq n$ is the index of a word replaced by its synonym. The parameter of this premise is the number of selected words k , formulated as $\theta = \theta(k)$.
- **Random Insertion:** The premise here is to randomly select words in the sentence x_{pre} that are neither stop words nor special tokens, and concatenate each of these words with the next word in one of its possible bi-grams, i.e., $x_{aug} = x_1x_2 \dots x_{k_i}xb_{k_i} \dots x_n$, where (x_{k_i}, xb_{k_i}) is a bi-gram of the word at position $k_i \leq n$. The parameter of this premise is the number of selected words k , formulated as $\theta = \theta(k)$.
- **Random Swap:** The premise here is to randomly select tuples of two words in the sentence x_{pre} and swap their positions where they are not consecutively restricted, i.e., $x_{aug} = x_1x_2 \dots x_{k_j} \dots x_{k_i} \dots x_n$, where $k_i < k_j \leq n$ are the indices of words whose position is interchanged. The parameter of this premise is the number of selected tuples k , formulated as $\theta = \theta(k)$.
- **Random Deletion:** The premise here is to randomly remove selected words with a certain probability in the sentence x_{pre} , i.e., $x_{aug} = x_1x_2 \dots x_{-k_i} \dots x_n$, where $k_i \leq n$ is the index of a word that will be deleted. The parameters of this premise are the number of chosen words k and the probability p of removing them, formulated as $\theta = \theta(k, p)$.

An advantage of EDA is its simplicity, where neither a pre-trained language model nor fully estimated density is required. However, synthesized examples generated via EDA are label-inconsistent, since none of the above methods are conditional on the original label y_{pre} from the pre-set. Noisy labeling is a challenge presented by many data augmentation studies when the context of synthetic examples makes their attributes far different from those of the original data. References [8,10] addressed this problem by introducing a post-process that utilizes a pre-trained model on the specific task to re-label synthetic data. However, for the purpose of comparison, this study accepts noisy labeling from EDA by labeling an augmented example by its original label, meaning that we set $y_{aug} = y_{pre}$.

3.2. Back Translation

Back translation is an unsupervised approach for text data augmentation [2,12] that refers to the scheme of translating an example x_A into an example x_B in another language, and then translating it back to obtain an augmented example \hat{x}_A . As an open vocabulary modeling approach, back translation can generate diverse paraphrases while preserving the semantics of the original sentences, leading to significant performance improvements in semantic reasoning. Supposed that we have a machine translation model from Language A (e.g., Korean), $p_{A \rightarrow B}(x_B|x_A)$, that translates a sentence x_A into x_B , and we also have a reverse translation model from x_B into \hat{x}_A , $p_{B \rightarrow A}(\hat{x}_A|x_B)$. We define our data augmentation generative model as $p_\theta(x_{aug}|x_{pre}) = \sum_{x_B \in TS} p_{B \rightarrow A}(\hat{x}_A|x_B)p_{A \rightarrow B}(x_B|x_{pre})$, where $x_B \in TS$ is all possible translated versions in English of x_{pre} in Korean, and x_{aug} is a synthetic example in Korean generated by back translation from x_B , while θ is parameterized by pre-trained weights from both translation models.

Although back translation is not likely to mislabel the synthetic example, it suffers other weaknesses, such as out-of-vocabulary terms. For word-level NMT models, the translation of non-existing words has been solved through backup dictionary lookups [20,21]. However, such techniques are impractical, since because of differences in the morphological synthesis between languages, there is no one-to-one mapping between the source and target vocabulary. In this study, we overcome the out-of-vocabulary problem by using a

pre-trained model on a large dictionary. In particular, we utilize multilingual translation based on mBART [22]. The set of pre-trained weights (<https://github.com/UKPLab/EasyNMT> (20 February 2022)) is fine-tuned from 50 languages, which comprises a parallel corpus (<https://sites.google.com/site/iwslt/evaluation2017/TED-tasks>) between Korean and English.

Table 2 shows examples of the two approaches. In this example, the SR approach replaces the subject of the sentence “bicycle” with “cycle” and the subject “man” with “child”, but does not change the context of the action “is riding”. In RI, the first example shows the insertion of “local” to support the subject “man”, but the second example does not make sense when adding “range” to the original sentence. As shown in the case of RD and RS, swapping and deleting words make sentences absurd and grammatically incorrect. Interestingly, in the case of BT, the context of the sentence remains unchanged, even though the grammar is legitimately changed.

Table 2. Examples of text data augmentation.

Methods	Augmented Example	Translation
Original	한 남자가 자전거를 타고 있다	A man is riding a bicycle
SR	Ex1: 한 남자가 사이클을 타고 있다 Ex2: 한 아드님이 자전거를 타고 있다	Ex1: A man is riding a cycle Ex2: A child is riding a bicycle
RI	Ex1: 한 국한 남자가 자전거를 타고 있다 Ex2: 한 남자가 자전거를 타고 범위 있다	Ex1: A local man is riding a bicycle Ex2: A man is riding a bicycle and has a range
RS	EX1: 한 남자를 자전거가 타고 있다 Ex2: 한 남자가 자전거 타고를 있다	Ex1: A bicycle is riding a man Ex2: A man riding a bicycle
RD	Ex1: 한 남자 자전거를 타고 있다 Ex2: 남자 자전거를 타고 있다	Ex1: A man riding a bicycle Ex2: Man is riding a bicycle
BT	한 남자가 자전거를 타는 것입니다	A man is riding a bicycle

4. Experiments and Results

We repeated each experiment five times and report the average performance to ensure randomness of the dataset. For each experiment, we randomly picked a subset from a training set, known as the pre-set, and then a synthetic set was constructed from the pre-set and merged with the original set to create an augmented set. Later, we fine-tuned a pre-trained model on the augmented set for a specific task. Figure 1 depicts our procedure.

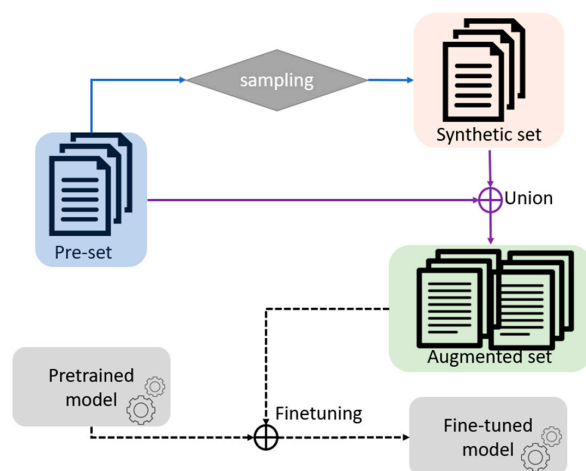


Figure 1. Prototype of the data augmentation process. To produce a fine-tuned model, we first used a pre-trained model and the pre-set (or set of premises) to learn the estimator for later constructing a synthetic set via a sampling process. The synthetic set was then united with the original dataset into an augmented set, which acted as the resource for fine-tuning the pre-trained model on a specific task.

4.1. Dataset

- **KorNLI:** This dataset is for the natural language inference task, consisting of 950,354 pairs of sentences translated from English. An example's label is one of three degrees of compatibility, entailment, contradiction, and neutral. The training set size is 942,854, while the development set has a size of 2490, and the test set has a size of 5010. In this study, we created random pre-sets by selecting 3927 samples (0.5%) from the training set.
- **KorSTS:** This dataset is for the semantic textual similarity task, consisting of 8628 pairs of sentences translated from English. An example's label ranges from 0 to 5, indicating the magnitude of similarity between two sentences. The training set size is 5749, while the development set has a size of 1500, and the test set has size 1379. In this study, we created random pre-sets by selecting 1725 samples (30%) from the training set.
- **NSCM:** This dataset is for the sentiment analysis task, consisting of 200 k movie reviews collected by Naver movies. All reviews are shorter than 140 characters and are classified into two categories (0: negative, 1: positive). The training set size is 150 k reviews, while the test set has 50 k reviews. In this study, we created random pre-sets by selecting 3000 sentences (2%) from the training set.
- **Question Pair:** This dataset is for the duplication-checking task, consisting of 15,000 questions translated from English and arranged as pairs of sentences. The examples are classified into two types (0: no duplicate, 1: duplicated). The training set size is 6136 pairs, while the development set has 682 pairs, and the test set has 758 pairs. In this study, we created random pre-sets by selecting 1840 samples (30%) from the training set.

4.2. Downstream Task Evaluation

The evaluation process was common to all tasks: We repeated the trials five times on five different random pre-sets, as depicted above. In each experiment, we applied text data augmentation on the pre-sets to generate the corresponding synthetic sets, then took the union of the pre-set and the synthetic set to build the augmented set. We applied cross-validation with a 20% portion as a validation set from the augmented set for fine-tuning. Finally, we assessed the performance of the fine-tuned model on the original test set of each task, and we report the average outcome across the five different experiments, as shown in Tables 3 and 4.

Table 3. Performance on downstream tasks of pre-trained ENLIPLV2: NLI, natural language inference (accuracy); STS, semantic textual similarity (Spearman).

Dataset	Full Training Dataset	Pre-Dataset	Augmented Dataset	
			EDA	Back Translation
KorNLI	83.21	71.27	73.93	72.81
KorSTS	84.75	81.70	81.54	81.99

Table 4. Performance on downstream tasks of pre-trained KoELECTRA-Base-v3: NLI, natural language inference (accuracy); QUAD, question duplication (F1); STS, semantic textual similarity (Spearman); NSCM, sentiment classification (accuracy).

Dataset	Full Training Dataset	Pre-Dataset	Augmented Dataset	
			EDA	Back Translation
KorNLI	82.24	71.25	73.49	72.13
Question Pair	95.25	94.19	94.85	93.13
KorSTS	85.53	81.67	82.54	83.86
NSCM	90.63	86.32	86.40	85.60

For the learning model, we utilized two Korean-monolingual pre-trained models from two different resources: ENLIPL-v2 and ELECTRA-Base-v3. The pretrained weights were originally obtained from the official release of each model. Hence, we only report the settings used in this study. We by-turn fine-tuned each learning model on the dataset of each specific task using the cross-entropy objective function, on 10 epochs with the Adam optimizer. The batch size was set to 32, and sentences with lengths less than 128 were padded with “[PAD] = 0” at their ends. The learning rate was warmed up within two epochs and kept unchanged at 5×10^{-5} .

To obtain the results shown in Tables 3 and 4, we generated synthetic data from a random pre-set using EDA and back translation. We created a new instance for each sample in the pre-set, which means that the number of samples generated was equal to the number of elements of the pre-set. For EDA settings, we randomly picked one of four transformations (SR, RI, RS, RD) with a replacement probability of 20%. In terms of implementation, we referred to the original implementation of EDA for English and Korean versions (<https://github.com/toriving/KoEDA> (21 February 2022)). In this way, our augmented set was twice the size of the pre-set. We also note that the EDA method runs remarkably faster than back translation: under 10 ms with EDA, compared to 1 s with back-translation.

For pre-trained models, we used ENLIPL-v2 (results shown in Table 3) and KoELECTRA-Base-v3 (results shown in Table 4). ENLIPL-v2 is the pre-trained language model based on dynamic masking from RoBERTa [23] and n-gram masking from ALBERT [24]. The model was trained on Korean Wikipedia and news in a total of 174M sentences, with mask language modeling (15% masking), next sentence prediction, and sentence order prediction objective. In this study, we used a large version of ENLIPL made up of 24 layers, with a hidden size of 1024 and 64 attention heads. KoELECTRA is a transformer-based pre-trained language model with self-supervised learning ELECTRA to distinguish “real” input tokens and “fake” input tokens generated by another network [25]. The model was trained on 20 GB of Korean text from various sources. For this study, we used version 3 of base KoELECTRA built from 12 layers of 768 embedding size and 12 attention heads.

We report the results for four language comprehension tasks: semantic textual similarity, natural language inference, sentiment analysis, and duplication verification. Data augmentation showed promising effectiveness for all tasks on both pre-trained models, where the performance was slightly better than that with no data augmentation. We argue that the improvement is not significant due to two issues. Firstly, EDA and BT struggle with keeping labels consistent since both of them are unsupervised models. Secondly, we only generated a small synthetic set, and the generated set was set fixed before fine-tuning, which reduced the diversity of generated samples.

Another remark is the inconsistency in the performance of BT and EDA concerning either task or model. In most cases, the EDA approach shows better performance than BT. We notice that there is not much difference in performance considering the choice of a pre-trained model. Therefore, we looked into the synthetic dataset and found that the BT approach lacks innovation; typically, when the sentence’s context is simple or universal, the synthesized sentence is completely identical to the original. In contrast, EDA can create a new instance for each pattern regardless of the context of the pattern. However, the sentences produced are sometimes grammatically incorrect or meaningless. Hand-crafted lexical data limit replacement-based methods such as SR and RI to not produce diverse patterns from the source text. Also, if used alone, random deletion and random swap can affect performance to the same extent as changing the context of a sentence without adding any of the equitable information.

5. Conclusions

This study provides a comprehensive investigation of data augmentation in the Korean language. We conducted various experiments with the current best pre-trained models and two well-known text data augmentation approaches. Based on our experiments and a

vast number of prior research efforts, we suggest that when fine-tuning a language model, either in Korean or in another language, data augmentation is a rule of thumb. In detail, our experimental results show that using text data augmentation can gain better performance on both language comprehension tasks (NLI: 2.24%, STS: 2.19%, QDV: 0.66%) and sentiment classification (STC: 0.08%). Nevertheless, ablation studies, as well as experiments with more sophisticated data generators, are necessary to adequately emphasize the advantages of text data augmentation, and we hence leave it as future work. We also note that this study is the first insight into data augmentation for the Korean language; these findings could be a potential starting point for further studies on natural language processing of Korean.

Author Contributions: Conceptualization, D.T.V. and G.Y.; methodology, D.T.V.; formal analysis, D.T.V.; investigation, D.T.V.; resources, D.T.V.; data curation, D.T.V.; writing—original draft preparation, D.T.V.; writing—review and editing, D.T.V. and J.K.; visualization, D.T.V.; supervision, J.K.; project administration, G.Y. and C.L.; funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Ministry of Culture, Sports, and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program (R2020060002).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: KorSTS and KorNLI can be found at <https://github.com/kakaobrain/KorNLUDatasets> (18 February 2022). The Naver Sentiment Analysis (NSCM) dataset can be found at <https://github.com/e9t/nsmc> (18 February 2022), and the Question Pair dataset can be found at https://github.com/songys/Question_pair (18 February 2022).

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

References

1. Connor, S.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60.
2. Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.T.; Le, Q.V. Unsupervised Data Augmentation for Consistency Training. *Adv. Neural Inf. Processing Syst.* **2020**, *33*, 6256–6268.
3. Johnson, M.; Schuster, M.; Le, Q.V.; Krikun, M.; Wu, Y.; Chen, Z.; Thorat, N.; Viégas, F.; Wattenberg, M.; Corrado, G.; et al. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 339–351. [[CrossRef](#)]
4. Wei, J.; Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv* **2019**, arXiv:1901.11196.
5. Kobayashi, S. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv* **2018**, arXiv:1805.06201.
6. Wu, X.; Lv, S.; Zang, L.; Han, J.; Hu, S. Conditional Bert contextual augmentation. In Proceedings of the International Conference on Computational Science, Faro, Portugal, 12–14 June 2019.
7. Kumar, V.; Choudhary, A.; Cho, E. Data augmentation using pre-trained transformer models. *arXiv* **2020**, arXiv:2003.02245.
8. Yang, Y.; Malaviya, C.; Fernandez, J.; Swayamdipta, S.; le Bras, R.; Wang, J.-P.; Bhagavatula, C.; Choi, Y.; Downey, D. Generative Data Augmentation for Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 1008–1025.
9. Schick, T.; Schütze, H. Generating Datasets with Pretrained Language Models. *arXiv* **2021**, arXiv:2104.07540.
10. Thakur, N.; Reimers, N.; Daxenberger, J.; Gurevych, I. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. *arXiv* **2020**, arXiv:2010.08240.
11. Schick, T.; Hinrich, S. Few-Shot Text Generation with Natural Language Instructions. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; Association for Computational Linguistics: Punta Cana, Dominican Republic, 2021; pp. 390–402.
12. Edunov, S.; Ott, M.; Auli, M.; Grangier, D. Understanding Back-Translation at Scale. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 489–500.
13. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
14. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv* **2019**, arXiv:1910.13461.

15. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv* **2019**, arXiv:1908.10084.
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
17. Reimers, N.; Gurevych, I. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv* **2020**, arXiv:2004.09813.
18. Cho, W.I.; Moon, S.; Song, Y. Open Korean Corpora: A Practical Report. In Proceedings of the Second Workshop for NLP Open Source Software (NLP-OSS); Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 85–93.
19. Lee, S.; Jang, H.; Baik, Y.; Park, S.; Shin, H. Kr-bert: A small-scale korean-specific language model. *arXiv* **2020**, arXiv:2008.03979.
20. Ham, J.; Choe, Y.J.; Park, K.; Choi, I.; Soh, H. KorNLI and KorSTS: New Benchmark Datasets for Korean Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 422–430.
21. Lee, H.; Yoon, J.; Hwang, B.; Joe, S.; Min, S.; Gwon, Y. Korealbert: Pretraining a lite bert model for korean language understanding. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 5551–5557.
22. Tang, Y.; Tran, C.; Li, X.; Chen, P.-J.; Goya, N.; Chaudhary, V.; Gu, J.; Fan, A. Multilingual translation with extensible multilingual pretraining and fine-tuning. *arXiv* **2020**, arXiv:2008.00401.
23. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
24. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
25. Clark, K.; Luong, M.-T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.