Iterative Translation-Based Data Augmentation Method for Text Classification Tasks

Sangwon Lee¹, Ling Liu², Wonik Choi¹
¹Department of Electrical and Computer Engineering, Inha University, Incheon, 22212, Republic of Korea
²College of Computing, Georgia Institute of Technology, Atlanta, GA, 30332, USA

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

This work was supported by the Korea Agency for Infrastructure Technology Advancement(KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 21BDAS-B158275-02). This work was supported in part by Institute of Information & Communications Technology Planning & Evaluation (IITP) (2020-0-01389, Artificial Intelligence Convergence Research Center (Inha University).

ABSTRACT A well-known limitation of existing rule-based text augmentation is that it cannot be applied to other languages because it depends on grammatical and structural characteristics. Moreover, most text Generative Adversarial Networks (GAN) are unstable in training due to inefficient generator optimization and rely on maximum likelihood pre-training. This paper addresses the above problems by proposing a novel augmentation method with a Sentence Generator (SG) and Sentence Discriminator (SD) for Iterative Translation-based Data Augmentation (ITDA). This paper makes three original contributions. First, the ITDA SG is designed to provide universal multiple-language support by generating comprehensive augmented sentences through serial and parallel iterations of an existing translator, such as Google Translate. Second, given that the quality of the generated sentences varies depending on the translation combination or the type of sentence, the ITDA addresses this issue using a discriminator to achieve sentence augmentation, which can select high-quality augmented data using a text classifier. Third, the ITDA can perform sentence augmentation for 109 different languages using discriminators based on text classifiers trained for a specific language or type of data set. Extensive experiments are conducted to evaluate the efficacy of the ITDA using a Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (BiLSTM), CNN-BiLSTM, and self-attention. The results demonstrate that when the ITDA is applied to 480 sentence classification tasks, the average accuracy increases by 4.24%.

INDEX TERMS Deep learning, data augmentation, natural language processing, text classification.

I. INTRODUCTION

Many organizations collect increasing amounts of data to build complex data analytics, machine learning, and Artificial Intelligence models [1]. Machine learning models trained on smaller data sets often do not perform well enough. Data augmentation methods can help train more robust models, even when training over smaller data sets [2] and are commonly used in computer vision [3] and speech [4]. The image data augmentation method has been used in many studies because it is universal and fast. In contrast, the text data augmentation methods exhibit strong language-dependent characteristics because text data are highly sensitive to added random noise or minor changes. Some previous studies address this limitation in text augmentation by focusing on rule-based methods that is highly language dependent [5, 6, 7]. Consequently, such data augmentation methods have limited utility for other languages. For example, these rule-based methods are intensively studied with English and lack general-purpose utility. The data augmentation method is more efficient for small data sets. Moreover, it is in high demand for other languages to use data augmentation to enhance language model training efficiency with small data sets.

However, universal data augmentation methods in Natural Language Processing (NLP) have not been fully explored because it is challenging to establish generalized rules for other languages [5]. Moreover, most text Generative Adversarial Networks (GAN) [8] are unstable in training (due to inefficient optimization of the generator) and rely on maximum likelihood pre-training [9].
This paper proposes a novel Iterative Translation-based Data Augmentation (ITDA) method that can be applied to multiple languages. Like GAN, the ITDA consists of a Sentence Generator (SG) and Sentence Discriminator (SD), but they are independently trained. The ITDA SG is designed to provide universal multi-language support by generating comprehensive augmented sentences through serial and parallel iterations of an existing translator such as Google Translate.

We address the problem that the SG may not always produce high-quality transformed sentences by optimizing the SG with an SD based on a text classifier, which learns to filter out low-quality sentences. For our first prototype, we develop a deep learning model using a Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) as the ITDA SD. The SD trains the source sentences with labels and predicts the label of the transformed sentences generated by the SG. If the predicted transformed sentence label differs from the source sentence label used to generate the transformed sentences, it is removed.

This method is effective in discriminating low-quality transformed sentences. The main advantage of this independent discriminator is that it can use any classifier model depending on the characteristics of the corresponding language. By intelligently combining an SG and SD, the ITDA can augment sentences in 109 languages supported by Google Translate using a text classifier best suited to the specific language and data set.

We verify the efficacy of the ITDA by conducting extensive sets of experiments using CNN, BiLSTM, CNN-BiLSTM, and self-attention on five real-world data sets in English and Korean. The results demonstrate that when the ITDA is applied to the 480 sentence classification tasks, the average accuracy increases by 4.24%. Furthermore, by changing the SG parameters \( (i, n) \), the ITDA improves performance in most cases and can be applied to languages with different grammar rules.

II. RELATED WORK

Easy Data Augmentation (EDA) [5] proposed four methods to augment sentence data using the grammatical features of English: Synonym Replacement (SR), Random Insertion (RI), Random Swap (RS), and Random Deletion (RD). The authors conducted three experiments. The first experiment augments 500, 2,000, and 5,000 original data sets with four EDA methods and compares the accuracy with Recurrent Neural Network (RNN) [10] and CNN [11] models. The second experiment uses five data sets and observes the performance improvement of each model using 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 (%) of the data. In the third experiment, the performance is compared using only one of each of the four methods. In all experiments, the smaller the amount of data, the higher the performance.

EDA has been used in many studies where data are scarce due to its fast processing time and simple implementation. However, some of the four methods may not function well when applied to languages with different grammar rules. Other studies have used data noising as smoothing [12] and predictive language models for SR [13]. Hierarchical Data Augmentation (HDA) augments data by extracting the most relevant parts from sentences and connecting them in order using a hierarchical attention network model [14]. Most sentence data augmentation research focuses on one language. The proposed ITDA is a data augmentation method based on iterative translation for supporting multiple languages.

Translation-based text augmentation methods use predominantly Neural Machine Translation (NMT) [15]. One study explores strategies for learning monolingual data without changing the encoder-decoder NMT architecture by pairing monolingual learning data with automatic back-translation [16]. Back-translation has been proven effective by extensive experiments conducted in [17]. QANET improved the performance of question-and-answer tasks by generating new data through reverse translation technology that translates English to French and back to English [18]. Unsupervised Data Augmentation (UDA) is a semi-supervised method of learning through data augmentation (including back-translation) that achieves state-of-the-art results on a wide variety of language and vision tasks [19].

We propose an SG using iterative translations based on serial and parallel back-translation. Then, the SD in the proposed ITDA selects high-quality transformed sentences generated from the SG.

III. BACKGROUND

Google’s Neural Machine Translation (GNMT) [20] is a deep learning model used by Google Translate that currently supports translations for 109 languages. GNMT follows the general sequence-to-sequence learning framework [21] with attention [22]. Figure 1 illustrates GNMT’s structure.

**FIGURE 1. Google’s Neural Machine Translation model structure [20]**

GNMT has three components: an encoder network, a decoder network, and an attention network. The encoder transforms a source sentence into a list of vectors, one vector per input symbol. Given this list of vectors, the decoder produces one symbol at a time until the special end-of-sentence symbol is produced. The encoder and decoder are connected through
an attention module, enabling the decoder to focus on different regions of the source sentence during decoding.

Let \( X = x_1, x_2, \ldots, x_M \) be the sequence of \( M \) symbols in the source sentence and \( Y = y_1, y_2, \ldots, y_N \) be the sequence of \( N \) symbols in the target sentence. The encoder is simply a function of the following form:

\[
x_1, x_2, \ldots, x_M = \text{EncoderRNN}(x_1, x_2, \ldots, x_M) \tag{1}
\]

In this equation, \( x_1, x_2, \ldots, x_M \) is a list of fixed-size vectors. The number of members in the list is the same as the number of symbols in the source sentence. With the chain rule, the conditional probability of the sequence \( P(Y | X) \) can be decomposed as:

\[
P(Y | X) = P(Y | x_1, x_2, \ldots, x_M) = \prod_{t=1}^{N} P(y_t | y_{0:t-1}, x_1, x_2, \ldots, x_M) \tag{2}
\]

The decoder is implemented as a combination of an RNN network and a softmax layer. The decoder RNN network produces a hidden state \( y_t \) for the next symbol to be predicted, which is then processed through the softmax layer to generate a probability distribution over candidate output symbols. Like [23], GNMT uses a deep-stacked Long Short-Term Memory (LSTM) [24] network for both the encoder and decoder RNNs.

The attention module is similar to that in [22]. Let \( y_{t-1} \) be the decoder-RNN output from the past decoding time step. Attention context \( a_t \) for the current time step is computed as follows:

\[
s_t = \text{AttentionFunction}(y_{t-1}, x_t) \quad \forall t, \ 1 \leq t \leq M
\]

\[
p_t = \exp(s_t) / \sum_{t=1}^{M} \exp(s_t) \quad \forall t, \ 1 \leq t \leq M
\]

\[
a_t = \sum_{t=1}^{M} p_t \cdot x_t
\]

Where the attention function in this implementation is a feed-forward network with one hidden layer.

IV. ITERATIVE TRANSLATION-BASED DATA AUGMENTATION METHOD

A. SENTENCE GENERATOR

The SG generates a sentence by transforming the source sentence using Google Translate [20], which supports 109 languages. As depicted in Figure 2, the source sentence is translated into other languages \( i \) times serially using Google Translate. The meaning of the other language is a language randomly selected from the other 108 languages supported by Google Translate. After completing the \( i \)-time serial translation, it is finally translated into the language of the source sentence. By iterating the above operation \( n \) times in parallel, \( n \) transformed sentences can be generated. This serial and parallel sentence generation method has the advantage that it can be easily applied to the 109 languages supported by Google Translate regardless of language characteristics.

SG cannot always generate high-quality transformed sentences. We first remove the transformed sentence if it is the same as the source sentence. However, the proportion of the duplicate sentences may be high. This problem can be solved by adjusting \( i \). If we increase \( i \), the serial SG is less likely to generate the duplicate sentences because the strength of the transformation increases. However, the high \( i \) value leads to trade-offs in generating sentences that differ significantly from the source sentences. If these over-translated sentences are used as train data, the performance of the model will decrease. We address this problem by using an SD to select and remove unnecessary transformed sentences for training.

B. SENTENCE DISCRIMINATOR

The SD is based on a classifier suitable for a data set. The SD only learns source sentences. After learning the source sentences, the SD predicts the label of the transformed sentence generated by the SG. If the predicted label of the transformed sentence differs from the label of the source sentence, it is discriminated as an incorrectly transformed sentence.

FIGURE 2. Serial and Parallel Sentence Generator
The structure of the SD consists of a parallel structure merging CNN and BiLSTM [25], as depicted in Figure 3. We select BiLSTM (used primarily for NLP) and a CNN because they exhibit high performance on various classification problems. The embeddings used for the input of the BiLSTM layer of the SD are Glove [26] in English and FastText [27] in Korean.

The input of the convolution layer is used by adding one dimension to the input of the BiLSTM layer. The input of the CNN-BiLSTM model consists of one or two sentences depending on the task. When using one sentence as input, it has the structure depicted in Figure 3. In contrast, when using two sentences as input, it adds the same layer before the concatenate layer in Figure 3 to concatenate the features from the two sentences. The output is the label classification of the input sentence. The SD effectively predicts the label of the transformed sentence generated by the SG and finds and removes the transformed sentence whose predicted label differs from the label of the source sentence.

C. STRUCTURE OF ITERATIVE TRANSLATION-BASED DATA AUGMENTATION METHOD

The ITDA is based on the SG and SD, as depicted in Figure 4. As described in Section IV-A, The SG translates one source sentence serially $i$ times and iterates it in parallel $n$ times to generate $n$ transformed sentences. If the transformed sentence is the same as the source sentence or another transformed sentence, it is removed. Then, as described in Section IV-B, the SD discriminates the incorrectly transformed sentences by predicting the labels of the transformed sentences with a classifier trained only on the source sentences. Finally, the discriminated sentences are removed, and the remaining transformed sentences are used as the augmented sentences.

V. EXPERIMENTS

In this section, we evaluate the effectiveness of the ITDA by conducting experiments using five data sets: four English data sets and one Korean data set. We evaluate the proposed ITDA on three tasks: (1) a single-sentence task using two English data sets, (2) an inference task using two English data sets, and (3) a Korean single-sentence task using one Korean data set. The single-sentence and inference tasks fix the parameters of SG ($i = 3$, $n = 5$) and test the results according to the change in the number of train data. In the Korean single-sentence task, we test how the number of train data and the variation in the parameters ($i$, $n$) of SG affect the results. In all cases, the test data are permanently fixed for each data set. Table 1 shows the size of train and test data sets and the types of tasks. We represent the results of all experiments in terms of the average accuracy of 10 trials. The performance of data augmentation is determined by the difference in the accuracy between with and without data augmentation.

| Corpus | Train | Test | Task       |
|--------|-------|------|------------|
| CoLA   | 8.5k  | 1k   | acceptability |
| SST-2  | 67k   | 1.8k | sentiment   |
| RTE    | 2.5k  | 3k   | NLI         |
| WNLI   | 634   | 146  | coreference/NLI |
| 3i4K   | 55k   | 6.1k | Intention identification |
A. SINGLE-SENTENCE TASK

Corpus of Linguistic Acceptability (CoLA). The CoLA [28] consists of English acceptability judgments drawn from books and journal articles on linguistic theory. Each example is a sequence of words annotated with whether it is grammatically correct. We use this as a task to classify whether it is a valid English sentence. Table 2 illustrates the accuracy according to the number of train data when data augmentation is and is not applied. Each parameter of EDA is set to 0.25, and five augmented sentences are created. In Table 2, ITDA outperforms ITDA model with NSD. As shown in Table 2, EDA increases the mean accuracy by 0.54%, while ITDA increases it by 1.68%. In all cases, ITDA outperforms ITDA-NSD, which shows the benefits of SD. Specifically, the average accuracy of ITDA is 3.74% higher than ITDA-NSD.

Stanford Sentiment Treebank v2 (SST-2). The SST-2 [29] consists of sentences from movie reviews and human annotations of their sentiment. The task is to predict the sentiment of a given sentence. We use the positive/negative class split and use only sentence-level labels. Table 3 presents the accuracy according to the number of train data when data augmentation is applied and when it is not.

EDA, ITDA-NSD, and ITDA decrease the average accuracy of the four base deep learning models by 0.53%, 2.25%, and 0.95%, respectively. In the case of the SST-2 dataset, data augmentation is not effective because of the large amount of data (SST-2 number of train data: 67k). Since SST-2 consists of sentences about movie reviews, it contains many proper nouns. These proper nouns negatively affect on the performance of ITDA because they cause over-translation. For this reason, ITDA shows worse performance than EDA in the SST-2 experiment. We observe that the more SG generates incorrectly transformed sentences, the more SD tends to be unable to completely discriminate transformed sentences.

| Base model | Augmentation method | Data Rate | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% | Avg |
|------------|---------------------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|-----|
| CNN        | NONE                | 52.30     | 53.35 | 53.45 | 54.47 | 57.54 | 73.57 | 76.95 | 78.06 | 78.44 | 79.44 | 65.76 |
|            | EDA                 | 52.53     | 55.20 | 55.15 | 54.49 | 56.55 | 69.51 | 73.71 | 75.72 | 76.85 | 76.72 | 64.65 |
|            | ITDA-NSD            | 53.07     | 54.23 | 54.42 | 54.38 | 56.19 | 72.12 | 76.27 | 78.28 | 78.70 | 79.05 | 65.77 |
|            | ITDA                | 56.92     | 57.22 | 59.29 | 59.91 | 62.57 | 76.57 | 78.81 | 80.16 | 80.65 | 80.99 | 76.13 |
| BiLSTM     | NONE                | 52.57     | 54.94 | 56.48 | 56.53 | 60.65 | 75.48 | 78.09 | 78.97 | 81.37 | 81.64 | 67.76 |
|            | EDA                 | 52.97     | 55.44 | 56.48 | 56.53 | 60.65 | 75.48 | 78.09 | 78.97 | 81.37 | 81.64 | 67.76 |
|            | ITDA-NSD            | 53.42     | 57.87 | 57.41 | 55.41 | 57.71 | 71.57 | 75.44 | 76.66 | 77.52 | 78.16 | 66.20 |
|            | ITDA                | 55.97     | 56.36 | 56.42 | 57.39 | 60.51 | 75.01 | 77.93 | 78.76 | 79.49 | 79.91 | 67.77 |
| CNN-BiLSTM | NONE                | 54.57     | 55.47 | 56.89 | 56.42 | 59.57 | 74.86 | 78.95 | 80.24 | 80.13 | 82.35 | 69.24 |
|            | EDA                 | 55.38     | 56.16 | 55.83 | 55.72 | 57.53 | 70.55 | 76.16 | 77.37 | 77.72 | 78.65 | 65.95 |
|            | ITDA-NSD            | 54.06     | 55.12 | 54.76 | 55.74 | 57.76 | 74.54 | 78.70 | 79.77 | 79.83 | 80.30 | 67.06 |
|            | ITDA                | 57.94     | 59.00 | 59.57 | 61.08 | 62.45 | 77.00 | 79.89 | 80.83 | 81.50 | 82.02 | 70.13 |
| Self-attention | NONE            | 55.78     | 57.20 | 59.86 | 58.48 | 59.89 | 74.46 | 79.65 | 80.31 | 82.65 | 80.31 | 68.88 |
|            | ITDA-NSD            | 55.25     | 56.91 | 56.87 | 56.43 | 57.15 | 73.17 | 77.58 | 79.52 | 80.09 | 80.49 | 67.34 |
|            | ITDA                | 56.24     | 57.47 | 56.93 | 57.78 | 58.90 | 75.49 | 79.01 | 80.88 | 81.49 | 82.29 | 68.57 |

TABLE 2. Accuracy by percentage of CoLA train data

TABLE 3. Accuracy by percentage of SST-2 train data
B. INFERENCE TASK

Recognizing Textual Entailment (RTE). The RTE [30] data sets are obtained from a series of annual textual entailment challenges. RTE is a combination of data from RTE1 [31], RTE2 [32], RTE3 [33], and RTE5 [34], which are constructed based on news and Wikipedia text. Therefore, many sentences in RTE include proper nouns. In addition, the RTE data set has overfitting problem because the size of the train data is smaller than that of the test data. RTE consists of pairs of sentence-1 and sentence-2. Thus, data augmentation can be applied in three ways: Augmented sentence-1 (ITDA-1), Augmented sentence-2 (ITDA-2), and Matched Aug-1 and Aug-2 (ITDA-M). As shown in Table 4, the performances of CNN and BiLSTM models in the data rate ranging from 40% to 60% decrease when ITDA is applied due to over-translation by proper nouns. However, the overall improvement of the average accuracy is 0.24%. The main reason for this performance improvement is that the positive effect of ITDA’s overfitting prevention for the RTE data set is more dominant than the negative effect caused by over-translation.

Winograd Natural Language Inference (WNLI). WNLI [30] is created based on the Winograd Schema Challenge [35] data set. It predicts whether the sentence-2 with the pronoun substituted is entailed by the original sentence (the sentence-1). As shown in Table 5, the accuracy decreases as the data rate increase, due to a serious overfitting. The main reason for this overfitting problem of the WNLI data set is that the size of WNLI is relatively small (634 sentence pairs), imbalanced in two classes (65% not entailment).

Since WNLI consists of sentence pairs like RTE, three augmentation methods (ITDA-1, ITDA-2 and ITDA-M) can also be applied. In the sentence pairs, the sentence-1 is the original sentence, and the sentence-2 is the corresponding sentence-1 substituted with a possible referent of the ambiguous pronoun. The average accuracy of ITDA-2, which applies the ITDA to only the sentence-2, increases by 14.9%, which is the average performance improvement in these experiments. This is because over-translation for the sentence-2 occurs relatively less.

**TABLE 4.** Accuracy by percentage of RTE train data

| Base model | Augmentation method | Data Rate | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% | Avg |
|------------|---------------------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| CNN        | NONE                |           | 45.69 | 45.61 | 46.35 | 48.59 | 48.73 | 48.97 | 48.78 | 48.94 | 49.12 | 49.94 | 48.07 |
|            | ITDA-1              |           | 46.67 | 47.69 | 48.55 | 47.92 | 48.04 | 48.41 | 48.39 | 49.53 | 49.67 | 50.10 | 48.49 |
|            | ITDA-2              |           | 47.84 | 48.53 | 47.99 | 47.35 | 48.63 | 48.75 | 49.45 | 50.35 | 50.80 | 50.29 | 48.99 |
|            | ITDA-M              |           | 47.46 | 48.13 | 47.95 | 48.18 | 48.34 | 48.68 | 48.34 | 50.98 | 50.36 | 50.48 | 48.89 |
| BiLSTM     | NONE                |           | 46.37 | 48.12 | 48.24 | 49.61 | 51.06 | 51.26 | 51.73 | 52.35 | 53.61 | 53.50 | 50.59 |
|            | ITDA-1              |           | 45.45 | 46.79 | 46.12 | 48.00 | 48.67 | 49.10 | 51.78 | 50.57 | 52.29 | 53.27 | 49.22 |
|            | ITDA-2              |           | 46.43 | 47.33 | 50.16 | 49.88 | 50.16 | 49.29 | 50.08 | 51.53 | 52.76 | 53.22 | 51.08 |
|            | ITDA-M              |           | 45.32 | 48.31 | 48.84 | 49.88 | 49.01 | 48.54 | 49.92 | 52.71 | 52.79 | 52.64 | 49.67 |

**TABLE 5.** Accuracy by percentage of WNLI train data

| Base model | Augmentation method | Data Rate | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% | Avg |
|------------|---------------------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| CNN        | NONE                |           | 50.62 | 44.92 | 28.92 | 25.85 | 22.92 | 21.69 | 15.69 | 13.31 | 12.62 | 10.23 | 24.67 |
|            | ITDA-1              |           | 43.08 | 46.77 | 48.77 | 42.15 | 45.23 | 43.38 | 40.62 | 35.69 | 36.00 | 37.67 | 41.83 |
|            | ITDA-2              |           | 43.38 | 45.08 | 46.62 | 44.46 | 41.23 | 43.08 | 39.85 | 41.38 | 38.42 | 42.15 | 42.22 |
|            | ITDA-M              |           | 50.15 | 47.23 | 43.38 | 40.46 | 42.62 | 40.31 | 40.62 | 36.62 | 34.15 | 35.69 | 41.12 |
| BiLSTM     | NONE                |           | 43.38 | 37.23 | 27.08 | 27.38 | 26.31 | 24.62 | 18.00 | 14.00 | 14.31 | 11.85 | 24.42 |
|            | ITDA-1              |           | 44.64 | 41.69 | 38.00 | 38.92 | 40.31 | 41.08 | 36.62 | 36.77 | 34.26 | 30.77 | 38.11 |
|            | ITDA-2              |           | 41.54 | 40.62 | 38.00 | 40.31 | 36.00 | 40.00 | 36.00 | 37.85 | 34.46 | 42.31 | 38.71 |
|            | ITDA-M              |           | 40.46 | 38.92 | 36.00 | 38.00 | 37.38 | 40.31 | 40.62 | 36.62 | 34.15 | 35.69 | 38.31 |
| Self-attention | NONE        |           | 46.46 | 38.15 | 25.54 | 24.92 | 23.69 | 21.38 | 17.85 | 11.08 | 10.77 | 13.08 | 23.29 |
|            | ITDA-1              |           | 43.85 | 42.92 | 41.69 | 39.23 | 40.92 | 40.31 | 35.38 | 32.15 | 32.46 | 32.00 | 38.09 |
|            | ITDA-2              |           | 42.77 | 43.08 | 41.23 | 39.85 | 41.23 | 39.23 | 34.46 | 33.54 | 32.00 | 42.31 | 38.73 |
|            | ITDA-M              |           | 42.77 | 41.38 | 39.92 | 37.23 | 40.46 | 37.54 | 36.46 | 30.15 | 31.08 | 33.69 | 36.57 |
|            | Self-attention      |           | 47.39 | 35.69 | 28.15 | 29.54 | 27.38 | 25.85 | 19.23 | 18.77 | 16.46 | 15.54 | 26.40 |
|            | ITDA-1              |           | 46.77 | 40.71 | 40.15 | 38.77 | 38.92 | 38.92 | 34.46 | 33.08 | 34.46 | 33.23 | 38.18 |
|            | ITDA-2              |           | 41.85 | 42.31 | 38.15 | 40.15 | 38.62 | 37.69 | 32.92 | 34.92 | 34.46 | 43.69 | 38.48 |
|            | ITDA-M              |           | 44.46 | 39.08 | 34.00 | 38.46 | 36.77 | 37.54 | 34.31 | 39.38 | 36.00 | 34.77 | 37.48 |
C. KOREAN SINGLE-SENTENCE TASK

Intonation-aided intention identification for Korean (3i4K). 3i4K [36] is used to classify Korean sentences into one of seven categories: fragment, statement, question, command, rhetorical question, rhetorical command, or intonation-dependent utterance. Two sets of experiments are conducted: (1) changing the number of translation iterations (i) and (2) changing the number of transformed sentences (n).

1) EXPERIMENTS CHANGING NUMBER OF TRANSLATION ITERATIONS (i)

As described in Section IV-A, the SG uses Google Translate to translate the source sentence into another language i times serially to generate one transformed sentence. If this operation is repeated n times in parallel, n transformed sentences are generated. This experiment measures the amount of augmented data generated when varying the parameter i. The parameter n is constant at 15, and 7,000 3i4K data (1,000 for each label) are used. Duplicate sentences among the generated transformed sentences are removed as described in Section IV-A.

Figure 5 illustrates how many transformed sentences, discriminated sentences, and augmented sentences are generated by varying i. If there are no duplicate sentences, 105,000 transformed sentences are generated. Because the number of duplicate sentences decreases as i increases, the number of transformed sentences increases. However, the growth declines and is likely to converge at approximately 90,000. Figure 5 also illustrates the number of discriminated sentences when the SD is applied to the transformed sentences. As i increases, the number of transformed sentences that are discriminated to interfere with learning also increases because the number of transformed sentences that damage the source sentences increases as i increases. Augmented sentences are data obtained by subtracting discriminated sentences from transformed sentences. If i exceeds 3, more than half of the transformed sentences are discriminated and removed, with less than half remaining as the final augmented sentences.

Figure 6 illustrates the results of training the model with the final augmented data after removing the discriminated data and measuring the accuracy. As shown in Figure 6, ITDA shows the best performance when i = 3. When the value of i is smaller than 3, it shows poor performance because the transformed sentences from SG are very similar to the source sentences. On the other hand, when the value of i is larger than 3, the performance of ITDA decreases due to over-translation.

VI. Conclusion

We propose a novel text data augmentation method using ITDA, with three original contributions. First, ITDA combines SG and SD to provide general-purpose text data augmentation. Second, the ITDA SG generates transformed sentences through serial and parallel iterations of the existing translator. Third, the ITDA SD uses a classifier trained with the source sentences to discriminate and remove incorrectly transformed sentences. ITDA is designed for comprehensive data augmentation for multiple languages. With Google Translate for the sentence generator’s translator, we can generate transformed sentences for 109 languages. We can also use any classifier available for that language as the SD.
We conduct extensive experiments using four base deep learning models on five data sets. We observe through experiments that proper nouns in sentences negatively affect the SG because they cause over-translation. In contrast, when our ITDA is applied to sentences in which parts of the source sentence are replaced with pronouns, the average accuracy increases by 14.9%. Our experiments on 480 sentence classification tasks with the ITDA demonstrate that accuracy improves on average by 4.24%.

REFERENCES

[1] M. Kantarcioglu and F. Shao, “Securing big data in the age of ai,” in 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA). IEEE, 2019, pp. 218–220.

[2] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” arXiv preprint arXiv:1712.04621, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[4] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, “Audio augmentation for speech recognition,” in Sixteenth annual conference of the international speech communication association, 2015.

[5] J. Wei and K. Zou, “Eda: Easy data augmentation techniques for boosting performance on text classification tasks,” arXiv preprint arXiv:1901.11196, 2019.

[6] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, “Is bert really robust? a strong baseline for natural language attack on text classification and entailment,” in Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 05, 2020, pp. 8018–8025.

[7] J. Min, R. T. McCoy, D. Das, E. Pitler, and T. Linzen, “Syntactic data augmentation increases robustness to inference heuristics,” arXiv preprint arXiv:2004.11999, 2020.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” Advances in neural information processing systems, vol. 27, 2014.

[9] F. Huang, J. Guan, P. Ke, Q. Guo, X. Zha, and M. Huang, “A text gan for language generation with non-autoregressive generator,” 2020.

[10] P. Liu, X. Qiu, and X. Huang, “Recurrent neural network for text classification with multi-task learning,” arXiv preprint arXiv:1605.05101, 2016.

[11] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in 2017 International Conference on Engineering and Technology (ICET). Ieee, 2017, pp. 1–6.

[12] Z. Xie, S. L Wang, J. Li, D. Lévy, A. Nic, D. Jurafsky, and A. Y. Ng, “Data noising as smoothing in neural network language models,” arXiv preprint arXiv:1703.02573, 2017.

[13] S. Kobayashi, “Contextual augmentation: Data augmentation by words with paradigmatic relations,” arXiv preprint arXiv:1805.06201, 2018.

[14] S. Yu, J. Yang, D. Liu, R. Li, Y. Zhang, and S. Zhao, “Hierarchical data augmentation and the application in text classification,” IEEE Access, vol. 7, pp. 185 476–185 485, 2019.

[15] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” arXiv preprint arXiv:1409.1259, 2014.

[16] R. Sennrich, B. Haddow, and A. Birch, “Improving neural machine translation models with monolingual data,” arXiv preprint arXiv:1511.06709, 2015.

[17] S. Edunov, M. Ott, M. Auli, and D. Grangier, “Understanding back-translation at scale,” arXiv preprint arXiv:1808.09381, 2018.

[18] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, “Qanet: Combining local convolution with global self-attention for reading comprehension,” arXiv preprint arXiv:1804.09541, 2018.
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131446, IEEE Access

[19] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, “Unsupervised data augmentation for consistency training,” arXiv preprint arXiv:1904.12848, 2019.

[20] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey et al., “Google’s neural machine translation system: Bridging the gap between human and machine translation,” arXiv preprint arXiv:1609.08144, 2016.

[21] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Advances in neural information processing systems, 2014, pp. 3104–3112.

[22] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.

[23] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, “Addressing the rare word problem in neural machine translation,” arXiv preprint arXiv:1410.8206, 2014.

[24] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] A. Graves, N. Jaitly, and A. Mohamed, “Hybrid speech recognition with deep bidirectional LSTM,” in 2013 IEEE workshop on automatic speech recognition and understanding. IEEE, 2013, pp. 273–278.

[26] I. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[27] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.

[28] A. Warstadt, A. Singh, and S. R. Bowman, “Neural network acceptability judgments,” Transactions of the Association for Computational Linguistics, vol. 7, pp. 625–641, 2019.

[29] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, “Parsing with compositional vector grammars,” in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2013, pp. 455–465.

[30] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “Glue: A multi-task benchmark and analysis platform for natural language understanding,” arXiv preprint arXiv:1804.07461, 2018.

[31] I. Dagan, O. Glickman, and B. Magnini, “The pascal recognising textual entailment challenge,” in Machine Learning Challenges Workshop. Springer, 2005, pp. 177–190.

[32] R. B. Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor, “The second pascal recognising textual entailment challenge,” in Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment, 2006.

[33] D. Giampiccolo, B. Magnini, I. Dagan, and W. B. Dolan, “The third pascal recognizing textual entailment challenge,” in Proceedings of the ACLPASCAL workshop on textual entailment and paraphrasing, 2007, pp. 1–9.

[34] L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo, “The fifth pascal recognizing textual entailment challenge,” in TAC, 2009.

[35] H. Levesque, E. Davis, and L. Morgenstern, “The winograd schema challenge,” in Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning, 2012.

[36] W. I. Cho, H. S. Lee, J. W. Yoon, S. M. Kim, and N. S. Kim, “Speech intention understanding in a head-final language: A disambiguation utilizing intonation-dependency,” arXiv preprint arXiv:1811.04231, 2018.