IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 16, 2021, accepted June 27, 2021, date of publication July 1, 2021, date of current version July 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3094023

Data Augmentation for Sentiment Analysis Using Sentence Compression-Based SeqGAN With Data Screening

JIAWEI LUO[1], MONDHER BOUAZIZI[2], (Member, IEEE), AND TOMOAKI OHTSUKI[2], (Senior Member, IEEE)
[1]Graduate School of Science and Technology, Keio University, Yokohama 223-8522, Japan
[2]Department of Information and Computer Science, Keio University, Yokohama 223-8522, Japan
Corresponding author: Jiawei Luo (luo@ohtsuki.ics.keio.jp)

ABSTRACT Sentiment analysis refers to the process of automatically identifying the emotions expressed by people. Its accuracy is highly dependent on the amount of training data. However, it takes time and cost for humans to collect a large number of data. Many research works used generative models to generate a large amount of data based on a small amount of data for sentiment analysis. However, training on long texts and inaccurate sentiment information that might be generated are two severe challenges. It is difficult to improve the sentiment analysis accuracy effectively. In this paper, we propose a novel data augmentation framework based on Sequence generative adversarial networks (SeqGAN) to improve the sentiment analysis accuracy when the dataset already has a certain amount of data and contains long texts. Penalty-based SeqGAN is used to generate high-quality and diversified text data. Long short-term memory (LSTM) networks with attention mechanisms are used to conduct sentence compression for the training data of SeqGAN. A sentiment dictionary is used to retain the sentiment words for compressed data. We also propose a data screening method to obtain more accurate data from the generated data. The results of the usability, novelty, and diversity of the generated data show that the proposed sentence compression method can help SeqGAN learn more information from the long text data. The data generated by the proposed framework improve the classification accuracy of four classifiers applied on two distinct text datasets.

INDEX TERMS Data augmentation, hate speech detection, long short-term memory, machine learning, sequence generative adversarial network, sentence compression, sentiment analysis.

I. INTRODUCTION

Sentiment analysis is an essential field in natural language processing (NLP) [1]. It can analyze people's sentiments through their articles. With the development of social media, more and more people can express their opinions online, which makes sentiment analysis increasingly important. As another application of text classification on social networks, hate speech detection can detect if a piece of text contains some hateful or offensive information. Hate speech refers to the speech that disparages people based on ethnicity, religion, disability, gender, caste, and sexual orientation [2]. On a related topic, machine learning has achieved high accuracy in sentiment analysis and hate speech detection. However, it requires a large amount of high-quality training data that are hard to be collected by human annotators. Data augmentation is a technique that can help generate a large amount of data from a small amount. The frequently-used data augmentation technique in NLP is very straightforward and relies on simply replacing words by their synonyms or deleting words here and there for every text [3]. However, this kind of approach is usually effective only when the original dataset is small, which means it usually has no effect when the original dataset is already large. Besides, the structure of the generated text by this kind of approach is almost unchanged. This leads to two undesirable consequences. First and foremost, this usually has minor improvement, if any, on the accuracy of classification. Besides, the classifier trained on these generated data tends to over-fit quite easily.

The associate editor coordinating the review of this manuscript and approving it for publication was Shen Yin.

Many research works have focused on using generative models to conduct data augmentation for text. Generative models can generate new artificial data (also referred to as fake data) by learning the probability distribution of training data. However, text generation is a challenging task. So far, generative models in NLP have failed to effectively enhance the accuracy of classifier by data augmentation [4], [5]. Generative adversarial networks (GAN) [6] is a generative model that can generate high-quality and diversified images for image data augmentation [7]. However, applying GAN in text generation is challenging because GAN works with continuous numerical data, whereas text is neither numerical nor continuous, but rather a discrete set of non-sorted words and characters. Some researchers used reinforcement learning to solve this problem and achieved great success. For instance, two of the most efficient approaches which use such technique are mask generative adversarial networks (MaskGAN) [8], sequence generative adversarial networks (SeqGAN) [9]. A SeqGAN-based data augmentation approach showed that SeqGAN could generate high-quality and diversified texts [5]. However, even SeqGAN could not effectively enhance the accuracy of the classifier by data augmentation. Learning and generating long text is still challenging. In other words, despite its potentials, SeqGAN cannot learn long texts efficiently. Another problem is that SeqGAN might occasionally generate data that contain incorrect sentiment information. When the SeqGAN trained on positive data is used to generate artificial data, we found that some generated data do not contain sentiment information, and some even show negative emotion.

In this paper, a data augmentation framework is proposed for text classification. The framework uses SeqGAN to generate artificial text data. To train SeqGAN better on long text data, LSTM networks with attention are used to conduct supervised learning sentence compression. This sentence compression model is trained to get the short compressed data that contains the main information. Since the simple sentence compression might lead to a decrease in the sentiment analysis accuracy, a sentiment dictionary is used to add sentiment words to the compressed data. This guarantees that the compressed data contain not only the main information but also any existing sentiment information. The proposed sentence compression can provide more easy-to-learn compressed data for SeqGAN. After data generation, we train a classifier on the original training data and use it to classify the generated data. The generated data classified into the correct class by the classifier will be retained as the final data. The final data are more helpful to improve the accuracy classification. In the experiments, the framework is evaluated on a sentiment analysis dataset and a hate speech dataset that contains many long data. We evaluate the usability, novelty, and diversity of the generated data to assess whether the proposed sentence compression method can really help SeqGAN learn more information from the compressed data. We build four baseline classifiers to evaluate whether the data generated by the proposed framework helps improve the sentiment analysis accuracy. The hate speech dataset is used to assess whether the framework can be used for the datasets in other fields of text classification.

The remainder of this paper goes as follows: In Section III, we introduce some related work and describe some of the challenges for our work. In Section IV, we introduce the proposed framework and how the experiments are conducted. In Section V, we describe the datasets used in our experiments and the experimental setup. In Section VI, we show and discuss the results of classification accuracy. We also evaluate the usability, novelty, and diversity of the generated data. Finally, in Section VII, we conclude this paper.

II. MOTIVATION

Text generation is different from text classification. The classification model in text classification tasks is highly likely to exceed the human level. However, the training data of a generative model is only a few thousand texts, and even humans cannot learn accurate grammatical rules or extract enough sentiment information from these texts. This leads that the data generated by the generative models usually has some grammatical mistakes and incorrect sentiment information. Thus, it is difficult for the generative model to reach the human level. The generative model can generate high-quality data by being trained on a large amount of data. However, in the data augmentation field, it is necessary to generate a large amount of data from a small amount of data. The insufficient training data leads that it is challenging to generate high-quality data. This is why so far, almost all generative models cannot effectively improve the accuracy of classifiers through data augmentation. In our work, we try to find a way to effectively use the generative model and a small amount of training data to generate some data that can really increase the classification accuracy.

III. RELATED WORK

The accuracy of sentiment analysis and hate speech detection is likely to be reduced due to insufficient training data. Many researchers focused on the data augmentation technique to solve this problem. A commonly used data augmentation technique in NLP is randomly replacing or deleting words. In [3], Wei *et al.* did data augmentation for text classification using a synonym dictionary. The amount of data is increased by randomly replacing, inserting, swapping, and deleting words in every text. However, it shows that when the number of original data is more than 2,000, the effect becomes worse. In particular, the random deletion is likely to make the accuracy reduced. The main reason is that these operations performed on words did not consider the sentiment information. Some sentimental or information-embedded words may be deleted or swapped, which swaps the sentiment class of data. They also showed that when the number of original data is more than 2,000, changing more than 10% of words for every data will make the accuracy reduced. This means that only very few words can be changed for every text. This limits the amount of diversity in the generated data, which,

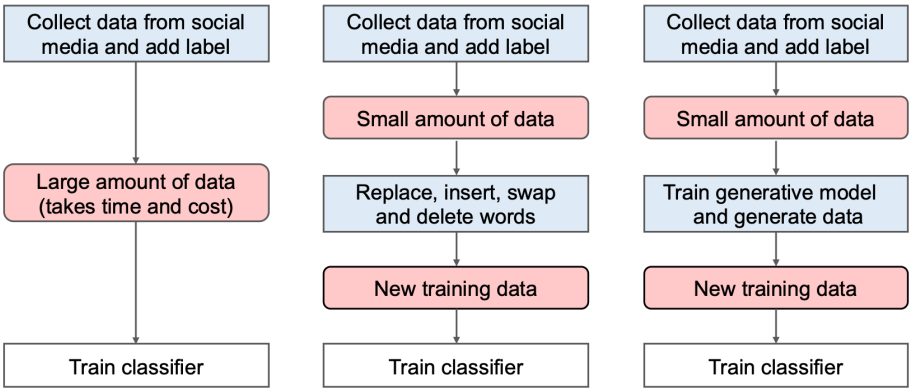


FIGURE 1. The methods to collect training data for a classifier. Left to right: general method, dictionary-based data augmentation, generative model-based data augmentation.

in consequence, leads to problems of over-fitting of the text classifiers.

GAN is a powerful generative model that shows a high ability to generate diversified images [7]. There are several variants of GAN in the field of NLP. In [4], Gupta *et al.* used conditional GAN (cGAN) [10] to perform data augmentation for sentiment analysis. In this model, the generator and discriminator are two feed-forward neural networks, which made the performance of this cGAN unsatisfactory. They pre-trained cGAN with a two-class sentiment analysis dataset that contains 1.6 million tweets to improve its data generation performance. The accuracy has been improved. However, it is not easy to find a large pre-training dataset for a new task when data augmentation is needed. When using this approach to expand a hate speech dataset, it is necessary to pre-train the cGAN with a large hate speech dataset.

SeqGAN is a generative model that applies reinforcement learning to the GAN's generator [9]. This algorithm can help GAN to deal with discrete words and generate grammatically accurate sentences. In [5], Wang *et al.* proposed a text generation framework SentiGAN that does not require pre-training with another dataset. SentiGAN is a generative model based on SeqGAN, and a penalty-based objective is proposed for the generators of SeqGAN to help it generate diversified and high-quality texts. This framework improved the sentiment analysis accuracy on four datasets. It outperformed several state-of-the-art text generation methods in terms of the quality and diversity of the generated texts. However, the work in [5] focused on learning and generating short texts. Only sentences that contain less than 15 words are selected to train SentiGAN. Learning and generating long texts is still challenging. The generated text's quality will be reduced, and the sentiment information may be lost when the generative model is trained on long texts. For a better understanding of the conventional data collection and data augmentation methods, the flowcharts of commonly used methods are shown in Fig. 1.

A sentence clustering-based approach is proposed to extract useful information from texts [11]. It showed the high performance in generating summarization from long texts, and it can solve the information overload problem. Information overload means the difficulty in understanding an issue, particularly from a long text with too much information. The generated short summarization will be easier to be analyzed, yet it still retains the main information of the original.

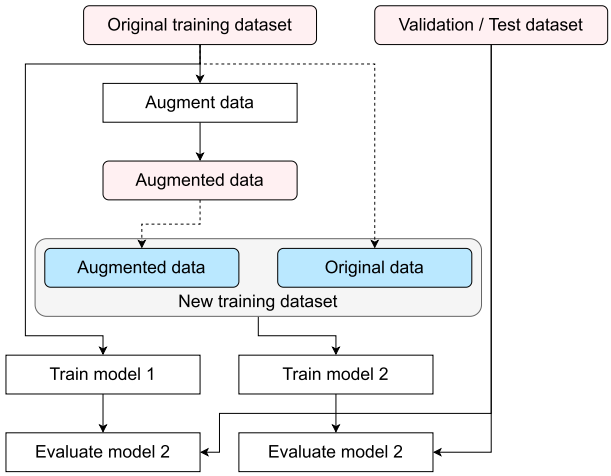


FIGURE 2. A flowchart of the proposed approach for data augmentation evaluation.

IV. PROPOSED METHOD

In this paper, a data augmentation framework is proposed to improve the accuracy of sentiment analysis. The proposed framework is also evaluated on a hate speech detection dataset to verify whether the proposed framework can be effectively applied to the datasets in other NLP fields.

In the proposed framework, a given training dataset, relatively small in size, is used to generate artificial data. The original training set is then combined with the augmented one to train a classifier, which we evaluate a validation or a test set. To evaluate the contribution of the newly generated data in improving the classification accuracy, we train another classifier on the original data solely, and we compare the two classifiers on the validation/test set. This is shown in Fig. 2.

Data augmentation, being the core contribution of this paper, is described in detail in this section. In Fig. 3, we show our proposed data augmentation framework. We briefly

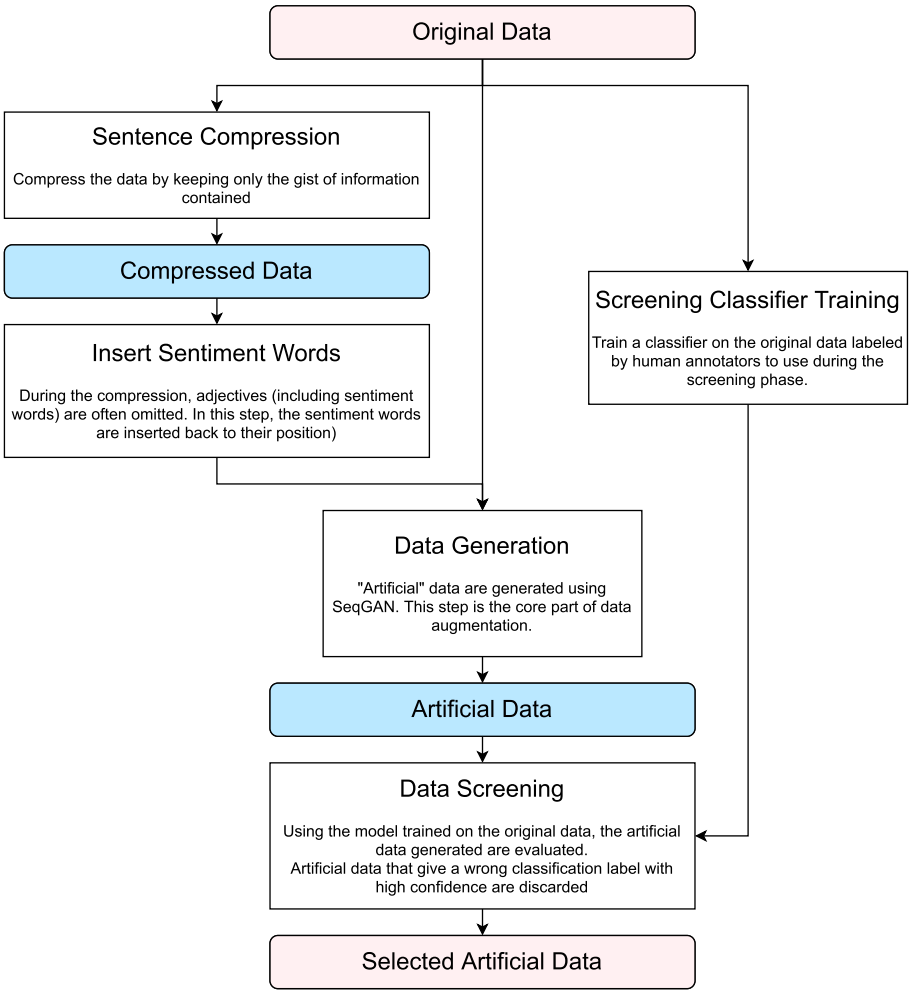


FIGURE 3. A flowchart of the proposed data augmentation.

describe here the different steps of the proposed approach. However, the remainder of this section describes in detail each step.

As described in the figure, we start by collecting a dataset (referred to as ''Original Data''). Each data instance from this dataset goes through a technique called sentence compression, whose objective is to reduce the size/length of the text. LSTM networks with attention mechanisms are used to do sentence compression on initial training data to get compressed data that can be easily parsed. Later, we explain why this is an important step that helps the generator create better quality artificial data. The compressed data are usually stripped from adjectives and adverbs which the compression algorithm considers as not important. However, for sentiment analysis tasks, these words (i.e., adjectives and adverbs) are usually very important and help identify the sentiment orientation of the text. Therefore, a sentiment dictionary is used to insert sentiment words that are lost during the sentence compression process, which allows retaining more sentiment information for compressed data. Afterwards, we use the original data and the compressed data to train SeqGAN and use it to generate artificial data. These data are not all useful, as they contain noisy data or data carrying wrong information. To identify the useful ones, we train a BiLSTM classifier on the original sentiment analysis training data and use it to classify the generated artificial data. This can help obtain the data that are more likely to carry the correct sentiment information. By doing so, we filter out the generated artificial data that are not very useful and can cause misclassification when used to train a new classifier. The final data are the generated data of the framework, which are used to improve the accuracy of sentiment analysis and hate speech detection.

A. SENTENCE COMPRESSION

SeqGAN has a good ability to generate text data. However, it still lags behind when learning information from long texts. Failure to learn all useful information from long texts leads to insufficiency or ineffectiveness in terms of information used for sentiment analysis. Sentence compression is a technology that can extract useful information from long texts. It can generate short texts that are easier to parse to solve the problem of information overload. Normally, long texts are directly used to train generative models, which leads to unsatisfactory results for many of them. Therefore, the framework's first

step is to provide SeqGAN with easy-to-learn short data by conducting sentence compression that considers sentiment words.

A supervised learning sentence compression model is used to do this. The model is implemented by the sequence to sequence (seq2seq) model [12]. The seq2seq model is typically composed of an encoder and a decoder. The encoder takes the source text as input, while the decoder generates a compressed sentence. LSTM networks were proposed to solve the problems of long-term dependencies and vanishing gradients. Thus, they can handle long texts much better than the conventional Recurrent Neural Networks (RNN) [13]. Therefore, LSTM cells [14] are used in the seq2seq model. A problem with LSTM is that it cannot give importance to some input words compared to others. This leads to a lower performance for memorizing important information in long sentences. The attention mechanism [15], [16] is used in the seq2seq model to solve this problem. Attention is a mechanism that can be embedded in networks and can dynamically select the attributes that relate to a given word in a given context. This mechanism is developed to increase the performance of the seq2seq model, it can help focus on the relevant regions of input and capture important semantic information.

The Gigaword dataset [17], which is a famous dataset in the sentence compression area, is used to train the compression model. There are roughly 3.8M training data in this dataset. Each data is a source-compression pair containing each article and its summary. The compression model updates its parameters to learn how to generate the summary from the long article when trained on the Gigaword dataset. After training, we input the initial training data that we want to expand into the compression model, and it will output compressed data.

To retain more sentiment information, we use the SentiStrength dictionary [18]. This dictionary contains a list of emotional words, where positive words have positive sentiment scores ranging from 1 to 5 and negative words have negative scores ranging from -1 to -5. This sentiment dictionary is used to check out whether sentiment words are lost during the sentence compression process. Sentiment words that exist in the original data and got deleted by the compression model are recorded. Out of all deleted sentiment words, the one with the greatest sentiment score (in absolute value) is inserted into the compressed data. The insert position of the sentiment word is determined by the words before and after the sentiment word in the original text. By doing this, we obtain the compressed data, which retain both the main information and the sentimental one.

B. DATA GENERATION

After sentence compression, the initial data and compressed data are used to train SeqGAN. SeqGAN is a generative model that applies GAN to NLP [9]. GAN is a generative model that can generate data by learning the probability distribution of training data. GAN is typically composed of

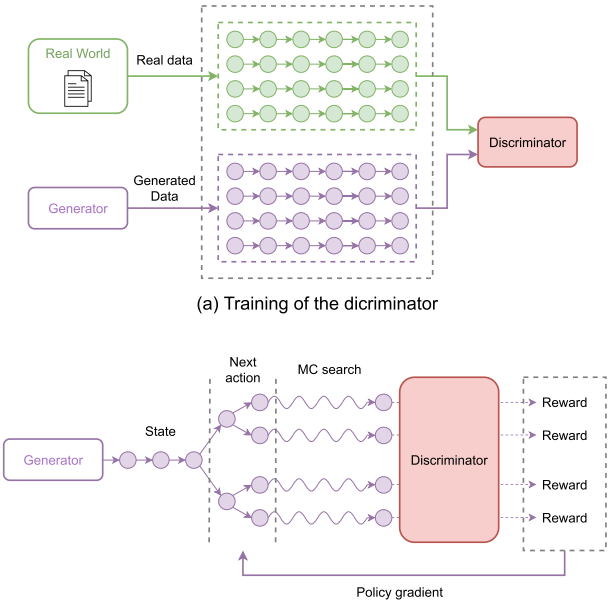


FIGURE 4. The architecture of SeqGAN: (a) shows the training of the discriminator with input data from the real world and Generator-generated data. (b) shows the training of the Generator with reinforcement learning.

two neural networks, which are the generator and the discriminator. The generator generates artificial data, while the discriminator classifies whether the data generated are real or not. GAN pits the two networks against each other until the generator creates data indistinguishable from the real ones provided by the training dataset, which means that the GAN can create data that look similar to the training data.

For a GAN based model, the generator and discriminator compete with the following value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))], \quad (1)$$

where G is a generative model and D is a discriminative model. $p_{data}(x)$ is the distribution of real data x, $p_z(z)$ is a prior distribution of noise variable z.

In Fig. 4, we show an illustration of the network architecture of SeqGAN as described in [9]. As can be seen, the training of the generator and the discriminator parts is done differently. The generator employs LSTM networks to generate the data, and the discriminator employs Convolutional Neural Networks (CNN) to discriminate real data from fake ones in SeqGAN. GAN, and neural networks in general, learn and generate continuous numerical data, whereas text is a discrete set of non-sorted words and characters. In other words, to train GAN, we need to either change the way data are generated or change the learning method. To solve this problem, SeqGAN applies reinforcement learning to its generator subnetwork, as shown in Fig. 4. Reinforcement learning gives positive rewards to the "good" generated candidates, and a negative reward/penalty to the "bad" ones. In such a way, the requirement for a continuous numeric sequence is bypassed.

In other words, this algorithm can help GAN deal with discrete words, and it formalizes the text generation problem as a sequential decision-making process. At each time step t, the generator produces sequences $s_t = \{x_1, \ldots, x_t\}$ for training, where x_t is a word token of a real word. Because the discriminator can only train on a complete sequence, the Monte Carlo search with roll-out policy [9] is used to sample the last words that are unknown and to be generated. The reinforcement learning makes the generator generate grammatically accurate data by forcing the generator to maximize reward $G(x|s;\theta_g)D(x;\theta_d)$, where θ_g, θ_d are the parameters of the generator G and the discriminator D, respectively. In [5], [19], a penalty-based objective is applied for the generator of SeqGAN. The penalty-based objective forces the generator to minimize the overall penalty $G(x|s;\theta_g)V(x)$, where $V(x) = 1-D(x;\theta_d)$. The generator's loss function of GAN, SeqGAN, and penalty objective-based SeqGAN are shown in the three formulas, respectively, as follows:

$$J_G(x) = \begin{cases} \mathbb{E}_{x\sim P_g}[-\log(D(x;\theta_d))] \\ \mathbb{E}_{x\sim P_g}[-\log(G(x|s;\theta_g)D(x;\theta_d))] \\ \mathbb{E}_{x\sim P_g}[G(x|s;\theta_g)V(x)]. \end{cases} \quad (2)$$

The penalty-based objective $G(x|s;\theta_g)V(x)$ can be thought of as adding $G(x|s;\theta_g)$ to the reward-based objective $G(x|s;\theta_g)D(x;\theta_d)$. The reward-based objective makes the generator generate grammatically accurate data and the penalty-based objective makes the generator generate more diverse data.

In the data augmentation field, the data with high diversity helps improve the classification accuracy compared with the data that have extremely accurate syntax. Thus the penalty objective-based SeqGAN is used to generate data in the framework. The initial training data and the compressed data are combined and used to train the generative model. After training, the generator can generate large amounts of artificial data with high diversity.

C. DATA SCREENING

The generative model can generate data limitlessly, but not all the data generated by the generative model are useful for training a text classifier. While training on the sentiment analysis dataset generates grammatically accurate sentences, these generated sentences are not necessarily accurate in terms of sentiment as well. Some of the generated data might not contain any sentiment-related information, and some might even contain the wrong ones. If we use these generated data as they are (i.e., with incorrect sentiment information) to train the sentiment analysis classifiers, the classification results might degrade.

Therefore, during this step referred to as ''data screening'', we train a classifier on the original dataset and run the classification on the generated data to obtain the sentimentally accurate instances. The classifier is implemented by BiLSTM networks. In the model, the first bi-directional hidden layer has 64 LSTM cells where the dropout is set to 0.4. This first layer is followed by a second bi-directional one with 32 LSTM cells, and a dropout equals 0.5. Afterward, the classification layer is a dense layer with two hidden units and a sigmoid activation function. The categorical cross-entropy loss function with the Adaptive Moment Estimation (Adam) optimizer is used. The training data is the initial training data. We pick 10% of these data and use them for validation. The training is stopped as soon as the model shows good performance. The classifier can predict the sentiment class of new data and output the probability of which class the current data is most likely to belong to. The classifier is used to classify the data generated by the proposed framework. The data whose sentiment class is most likely to be accurate is selected as the framework's final output data.

V. EXPERIMENTS

A. DATASETS

The data augmentation is conducted on a sentiment analysis dataset and a hate speech detection dataset by the proposed framework. Training on long texts is a big challenge for the generative model. [5] achieved the best results in the field of data augmentation for sentiment analysis. However, they experimented only on short data (length <= 15 words). To evaluate the framework on long data, we experiment on the data with a maximum length of 40 words. The two datasets used in our experiments are described in more detail below.

1) STANFORD SENTIMENT TREEBANK (SST) DATASET

The SST dataset is a movie review dataset publicly available [20]. It has two sentiment classes, which are ''positive'' and ''negative''. This dataset has a total of 6,920 training texts and 1,821 test texts. We randomly select 2,000 positive data and 2,000 negative data as the initial SST training dataset for the proposed framework. The 1,821 test data are used as the test dataset. The average number of words of the selected SST training data is approximately 20.

2) HATE SPEECH (HS) DATASET

The HS dataset is a tweets dataset collected by querying Twitter API [21]. It has three classes, which are ''hate'', ''offensive'', and ''clean''. The training dataset contains 21,000 tweets, and each class has 7,000 tweets. The test dataset contains 4,020 tweets, and each class has 1,340 tweets. We randomly selected 2,000 hate data, 2,000 offensive data, and 2,000 clean data as the initial HS training dataset for the proposed framework. The 4,020 test data are used as the test dataset. The average number of words of the selected HS training data is approximately 17.

B. EXPERIMENTAL SETUP

We do experiments on two datasets. For each class, the 2,000 selected data are used as the initial data of the proposed data augmentation framework. The experimental details are described below:

To evaluate the performance of the proposed sentence compression approach in the framework, we first train SeqGAN with 2,000 initial training data (represented by ① in the following) and generate 50,000 artificial data for each class.

2,000 artificial data (represented by ② in the following) that are different from training data are randomly selected. Next, the proposed sentence compression method is used to process ① and generate 2,000 compressed data (represented by ③ in the following). We use ①③ to train SeqGAN and generate artificial data. We randomly select another set of 2,000 artificial data (represented by ④ in the following) that are different from the training data ①③. Finally, we train the BiLSTM classifier on the 4,000 SST training data and 6,000 HS training data to conduct data screening, respectively. To evaluate the performance of data screening, we use the classifier to do classification on the generated data and conduct sentiment prediction. For each class, the BiLSTM classifier is used to select 2,000 artificial data (represented by ⑤ in the following) generated by SeqGAN without sentence compression and select another 2,000 artificial data (represented by ⑥ in the following) generated by SeqGAN with sentence compression. It is worth noting that every time we generate some data with SeqGAN, several generated data are identical to some data in the training dataset, and these are discarded. Therefore, the artificial data are all different from the original one.

For a more in-depth validation of the performance of our proposed framework, we compare it with a frequently-used data augmentation approach: EDA (Easy data augmentation) model [3] is used to conduct data augmentation on two datasets, respectively. The EDA approach randomly replaces, inserts, swaps, and deletes words to expand the dataset. There are two parameters in this approach: the first parameter is referred to as the α parameter. It defines the proportion of words in the sentence that need to be changed by each augmentation. The second parameter is referred to as the n parameter. It is used to describe the number of augmented sentences generated for each original sentence. α is set to 0.1 followed by their paper and n is set to 2, 3, 4, respectively. We evaluate this approach to compare it with our proposed framework. We will refer to the different variants of EDA with 2, 3, and 4 words to be changed as EDA_2, EDA_3 and EDA_4. As we will see later, this method has led to a very limited improvement in accuracy. We also explain the limits of the conventional approach [3] and why this method leads to a negligible improvement in the accuracy rather than an enhancement.

C. CLASSIFICATION METHODS

To evaluate whether the data generated by the proposed framework helps improve the accuracy of sentiment analysis and hate speech detection, we built four classifiers from the literature and conducted the classification on the aforementioned datasets, and compared the accuracy obtained with and without the use of the augmented data. The implemented classifiers include Logistic Regression (LR), Support Vector Machine (SVM), CNN, and LSTM [1], [2], [22]. These classifiers have shown great potential in text classification for sentiment analysis [22], hate speech detection [21], etc.

In particular, the LR classifier used is a typical L1-regularized Logistic regression classifier where the regularization parameter C is set to 1. SVM classifier is a powerful classification algorithm. The implementation of this classifier used here is a linear-SVM with L2 penalty parameters, and the regularization parameter is 1. The CNN has a 1D convolutional layer with 16 filters whose size is equal to 3, a global 1D max pool layer, a dense layer with two hidden units with a softmax activation function. The sparse categorical cross-entropy loss function with the Root Mean Square Propagation (RMDProp) optimizer is used for the CNN. In LSTM, the hidden layer has 15 LSTM cells where the dropout is set to 0.5. The dense layer has two hidden units with a softmax activation. The sparse categorical cross-entropy loss function with the Adam optimizer is used. For the input data of the classifiers, GloVe (Global Vectors) [23] is used to create word embeddings.

After performing the data augmentation, the data of each class of the same dataset are combined to evaluate the classification accuracy. We used ①, ①③, ①②, ①③④, ①⑤ and ①③⑥ as training data, respectively to train the four classifiers. Finally, the classification was performed on the original test data of two datasets.

VI. RESULTS

A. CLASSIFICATION ACCURACY

The classification accuracy on the sentiment analysis dataset and hate speech dataset are presented in Table 1 and Table 2. "SC" refers to the proposed sentence compression method, "SeqGAN" refers to the data generation by SeqGAN, "DS" refers to the proposed data screening, and "SC+SeqGAN+DS" is the proposed framework. The accuracy in the two tables is the classification results of four baseline classifiers and the average values. To better observe the effect of each part of the proposed framework, we also show the results when each part is used alone and in combination. Fig. 5 shows the changes in the classification accuracy on two datasets. The results are the changes in the average classification accuracy of the four classifiers. We can observe the improvement of accuracy after using each method.

After using EDA [3], the accuracy of the SST dataset increased by 0.2% when the dataset was expanded three times, and the accuracy of the HS dataset increased by 0.2% when the dataset was expanded four times. However, the accuracy is reduced in other situations. The paper of the EDA approach shows that for a dataset with a size of about 5,000, expanding the dataset to four times will get the best results. That is not the case in our experiments though. The main reason is that the size of two datasets is already large enough (SST = 4,000, HS = 6,000), which leads to the poor performance of EDA. They only used CNN, LSTM as a baseline classifier in their work [3], whereas, in our work, LR and SVM are used as well to evaluate the accuracy. Nevertheless, an important factor to keep in mind is that switching words with their synonyms usually makes the generated sentences not so different from the original ones. This leads, during

TABLE 1. The accuracy of four classifiers on the sentiment analysis dataset (%).

Methods	LR	SVM	CNN	LSTM	Average
SST dataset	75.1	78.5	81.1	81.8	79.1
EDA_2	74.7	78.1	81.1	81.7	78.9
EDA_3	74.8	79.0	82.0	81.5	79.3
EDA_4	75.0	77.4	82.0	81.5	79.0
SC	75.4	78.6	81.6	81.7	79.3
SeqGAN	75.7	78.9	81.3	81.4	79.3
SC+SeqGAN	76.0	79.0	82.7	82.3	80.0
SeqGAN+DS	75.6	79.1	82.3	82.0	79.7
SC+SeqGAN+DS	76.7	79.0	82.6	82.1	80.1

TABLE 2. The accuracy of four classifiers on the hate speech dataset (%).

Methods	LR	SVM	CNN	LSTM	Average
HS dataset	67.1	69.1	81.1	81.5	74.7
EDA_2	67.7	68.5	80.7	81.1	74.5
EDA_3	67.7	69.1	79.9	81.3	74.5
EDA_4	68.1	69.3	81.1	81.1	74.9
SC	68.1	69.6	80.5	81.5	74.9
SeqGAN	67.2	69.6	80.8	81.3	74.7
SC+SeqGAN	68.7	70.1	81.2	81.7	75.4
SeqGAN+DS	66.8	69.9	81.4	81.7	75.0
SC+SeqGAN+DS	68.0	69.9	82.0	82.5	75.6

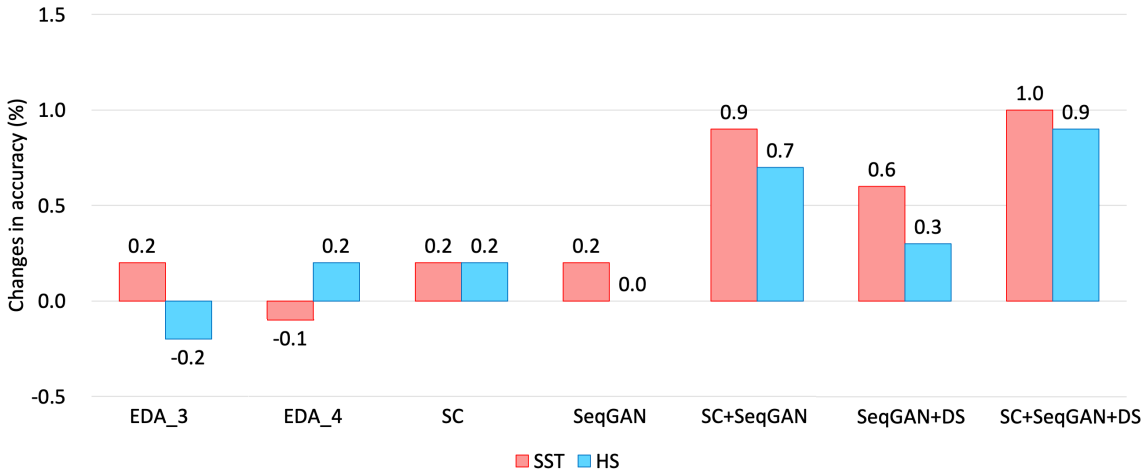


FIGURE 5. The changes in the classification accuracy compared to the real data on two datasets (%).

training, to fast overfitting to the training data, as a certain pattern in the training set will be observed frequently, leading the classifier to relate that particular pattern to the class quite often. Overall, EDA does not effectively improve their accuracy.

The classification accuracy is increased a little by a single use of the proposed sentence compression or SeqGAN. After combining the proposed sentence compression and SeqGAN to generate data, the accuracy is 0.9%, 0.7% higher than that of real data on the SST dataset and the HS dataset, respectively. This indicates that the data generated by the sentence compression is beneficial to train SeqGAN. Thus, SeqGAN can generate more meaningful training data for classifiers. By comparing "SeqGAN" and "SeqGAN+DS", we can see that after using data screening to process the generated data, the classification accuracy is improved by 0.4% and 0.3% on the SST dataset and the HS dataset, respectively. This shows that the proposed data screening helps obtain more accurate data from the generated data. After using sentence compression, SeqGAN, and data screening in combination, the accuracy is increased by only 0.1% and 0.2% compared with using sentence compression and SeqGAN. This may be because the SeqGAN trained with the compressed data can generate sufficiently accurate data. In this situation, the BiLSTM classifier cannot make a more precise classification on the generated data. Compared with the classification accuracy on the initial real data, the proposed framework improves the accuracy by 1.0% and 0.9% on the SST dataset and the HS dataset, respectively. This shows that the proposed framework can improve not only the accuracy of the two-class sentiment analysis dataset but also the accuracy of the three-class hate speech dataset.

B. USABILITY AND NOVELTY OF THE GENERATED DATA

SeqGAN is reported to have a good ability to generate more novel data, particularly after using a penalty-based objective for the generators [5]. To evaluate whether the proposed sentence compression approach is substantially helpful to the

TABLE 3. Usability of the Generated Data (%).

Methods	SST	HS
SeqGAN	13.7	27.3
SC+SeqGAN	31.6	58.5

TABLE 4. Novelty of the Generated Data (%).

Methods	SST	HS
SeqGAN	48.9	65.4
SC+SeqGAN	56.2	67.6

data generation of SeqGAN, the usability and novelty of the generated data were evaluated.

The usability means how much the percentage of data left after deleting the repetitive data is. In the experiment, we generate 50,000 artificial data for each class and count the repeated ones to calculate the usability. It shows the generation efficiency of the generative model. The novelty means how different the generated data and training data are. The novelty of the generated data is calculated by the Jaccard similarity. Given two sets, the Jaccard similarity is the ratio between the size of their intersection and the size of their union. The Jaccard similarity of two generated data d_1, d_2 is calculated as follow:

$$Sim(d_1, d_2) = \frac{|d_1 \cap d_2|}{|d_1 \cup d_2|}. \tag{3}$$

We calculate the maximum similarity between each generated data d_i and real data r_j in the dataset. The opposite of the maximum similarity is considered as the novelty of each data:

$$Novelty(d_i) = 1 - \max\{Sim(d_i, r_j)\} \quad i \in \{1, 2, \cdots n\}, j \in \{1, 2, \cdots m\}, \tag{4}$$

where n is the total amount of the generated data, m is that of the real data. The novelty shows whether the generative model can independently generate new data rather than repeatedly generate the same instances of the original training data.

The results of the usability and novelty of the generated data are given in Table 3 and Table 4, respectively. The usability is 13.7% and 27.3% on two datasets when the data are generated by a single use of SeqGAN, which indicates that most of the generated data are repeated. The low percentage of the single-use of SeqGAN is mainly because learning long data is difficult. When SeqGAN learns long text data, it can hardly learn all the information in data, particularly the long-term dependency relationships. As a result, the generative model learns less information and generates existing data to avoid generating the wrong ones. The usability improved by 17.9% and 31.2% on two datasets after using compressed data to train SeqGAN. The novelty is also improved by 7.3% and 2.2% on two datasets. This shows that the proposed sentence compression can provide SeqGAN with more easy-to-learn short texts. Thus, SeqGAN can learn more information from training data and generate more novel data by itself.

C. DIVERSITY OF THE GENERATED DATA

The training data with high diversity is significant for improving the classification accuracy of text classifiers. To evaluate whether the proposed framework can generate various data, the generated data's diversity is assessed by the Jaccard similarity. The maximum similarity between each data d_i and other data d_j in the dataset is calculated. The opposite of the maximum similarity is considered as the diversity of each data:

$$Diversity(d_i) = 1 - \max\{Sim(d_i, d_j)\} \quad i, j \in \{1, 2, \cdots n\}, j \neq i, \tag{5}$$

where n is the total amount of the generated data.

TABLE 5. Diversity of the Real Data and the Generated Data (%).

Methods	SST	HS
Real Data	70.6	76.7
EDA	9.0	9.5
SeqGAN	60.7	64.5
SC+SeqGAN	64.7	70.5

The average values of diversity are shown in Table 5, "Real Data" refers to the initial data of the data augmentation methods. The diversity of the data generated by the proposed framework is improved by 55.7% on the SST dataset and 61.0% on the HS dataset compared with EDA. The low diversity of the data generated by EDA is because it changes the data structure a little. After using the proposed framework, the diversity of the generated data is improved by 4% on the SST dataset and 5.5% on the HS dataset compared with SeqGAN. This shows that the proposed sentence compression method can help SeqGAN learn more information from the compressed data and generate more diverse data.

VII. CONCLUSION

In this paper, a data augmentation framework that combines supervised learning sentence compression, SeqGAN, and data screening was proposed. In the proposed framework, SeqGAN is used to generate text data to solve the insufficient diversity problem of commonly used methods. Since the long text training is a big challenge for SeqGAN, a sentence compression method is proposed. During the sentence compression process, the sentiment words are retained to keep more sentiment information. The proposed data screening approach can delete the generated data that contain incorrect sentiment information.

The framework is evaluated on the sentiment analysis dataset and the hate speech dataset that have a relatively large amount of data out of which, many are actually long. The proposed data augmentation framework outperforms the state-of-the-art data augmentation method in the quality, usability, novelty, and diversity of the generated data. After using the proposed sentence compression, the usability is improved by 24.6%, and the novelty is improved by 4.8% on average. The diversity of the generated data of the proposed framework is improved by an average of 58.4% compared with the conventional method EDA. The data generated by the proposed framework improve the classification accuracy by 1% on some of the benchmark sentiment analysis dataset available. The proposed framework is a novel data augmentation

method that can truly improve the accuracy of the actual sentiment analysis dataset without additional conditions. The performance on the hate speech dataset shows that the framework can be used for the datasets in other fields of text classification.

REFERENCES

[1] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowl.-Based Syst.*, vol. 89, pp. 14–46, Nov. 2015.

[2] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in *Proc. 5th Int. Workshop Natural Lang. Process. Social Media*, 2017, pp. 1–10.

[3] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 6382–6388.

[4] R. Gupta, "Data augmentation for low resource sentiment analysis using generative adversarial networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 7380–7384.

[5] K. Wang and X. Wan, "SentiGAN: Generating sentimental texts via mixture adversarial networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4446–4452.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process Syst.*, 2014, pp. 2672–2680.

[7] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[8] W. Fedus, I. Goodfellow, and A. M. Dai, "MaskGAN: Better text generation via filling in the," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.

[9] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence generative adversarial nets with policy gradient," in *Proc. 31th AAAI Conf. Artif. Intell.*, 2016, pp. 2852–2858.

[10] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[11] P.-Y. Zhang and C.-H. Li, "Automatic text summarization based on sentences clustering and extraction," in *Proc. 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol.*, 2009, pp. 167–170.

[12] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst*, 2014, pp. 3104–3112.

[13] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2873–2879.

[14] S. Sakti, F. Ilham, G. Neubig, T. Toda, A. Purwarianti, and S. Nakamura, "Incremental sentence compression using LSTM recurrent networks," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2015, pp. 252–258.

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

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[17] G. David and C. Cieri, *English Gigaword LDC2003T05*. Philadelphia, PA, USA: Linguistic Data Consortium, 2003. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2003T05

[18] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social Web," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 63, no. 1, pp. 163–173, Jan. 2012.

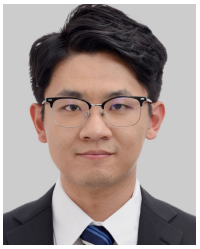
[19] M. Duan and Y. Li, "Penalty-based sequence generative adversarial networks with enhanced transformer for text generation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–6.

[20] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct. 2013, pp. 1631–1642.

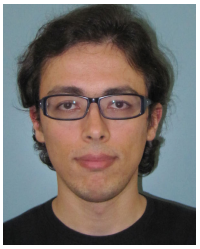
[21] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE Access*, vol. 6, pp. 13825–13835, 2018.

[22] M. Bouazizi and T. Ohtsuki, "A pattern-based approach for multi-class sentiment analysis in Twitter," *IEEE Access*, vol. 5, pp. 20617–20639, 2017.

[23] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.



JIAWEI LUO was born in Beijing, China, in 1995. He received the B.E. degree from the School of Software, Dalian University of Technology, Dalian, China, in 2018, and the double B.E. degree from the College of Information Science and Engineering, Ritsumeikan University, Kyoto, Japan, in 2018. He is currently pursuing the M.E. degree with the Graduate School of Science and Technology, Keio University, Tokyo, Japan. He is a member of IEICE.



MONDHER BOUAZIZI (Member, IEEE) received the Bachelor of Engineering (diploma) degree in communications from SUPCOM, Carthage University, Tunisia, in 2010, and the master's and Ph.D. degrees from Keio University, in 2017 and 2019, respectively. He worked as a Telecommunication Engineer (access network quality and optimization) for three years with Ooredoo Tunisia (Ex. Tunisiana). In 2015, he enrolled at Keio University, where he is currently working as a Special Assistant Professor.

TOMOAKI OHTSUKI (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Tokyo, Japan, in 1990, 1992, and 1994, respectively. From 1994 to 1995, he was a Postdoctoral Fellow and Visiting Researcher in electrical engineering at Keio University. From 1993 to 1995, he was a Special Researcher of Fellowships of the Japan Society for the Promotion of Science for Japanese Junior Scientists. From 1995 to 2005, he was with the Science University of Tokyo. In 2005, he joined Keio University, where he is currently a Professor. From 1998 to 1999, he was with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. He has published more than 180 journal articles and 400 international conference papers. He is engaged in research on wireless communications, optical communications, signal processing, and information theory. He is a fellow of the IEICE. He was a recipient of the Inoue Research Award for Young Scientist, in 1997, the Hiroshi Ando Memorial Young Engineering Award, in 1997, the Ericsson Young Scientist Award, in 2000, the Funai Information and Science Award for Young Scientist, in 2002, the IEEE 1st Asia-Pacific Young Researcher Award, in 2001, the 5th International Communication Foundation (ICF) Research Award, in 2011, the IEEE SPCE Outstanding Service Award, the 27th TELECOM System Technology Award, the ETRI Journal's 2012 Best Reviewer Award, and the 9th International Conference on Communications and Networking in China 2014 (CHINACOM '14) Best Paper Award. He served as the Chair for IEEE Communications Society and Signal Processing for Communications and Electronics Technical Committee. He has served as the General Co-Chair, Symposium Co-Chair, and TPC Co-Chair for many conferences, including IEEE GLOBECOM 2008, SPC, IEEE ICC2011, CTS, IEEE GCOM2012, SPC, IEEE ICC2020, SPC, IEEE APWCS, IEEE SPAWC, and IEEE VTC. He gave tutorials and keynote speech at many international conferences, including IEEE VTC and IEEE PIMRC. He was the Vice President and President of Communications Society of the IEICE. He served as a Technical Editor for *IEEE Wireless Communications Magazine* and an Editor for *Physical Communications* (Elsevier). He is currently serving as an Area Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and an Editor for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.