Human-centric Computing and Information Sciences

RESEARCH Open Access

Newspaper article-based agent control in smart city simulations

Euhee Kim[1], Sejun Jang[2], Shuyu Li[2] and Yunsick Sung[2*]

*Correspondence:
sung@dongguk.edu
[2] Department of Multimedia Engineering, Dongguk University-Seoul, Seoul 04620, Republic of Korea
Full list of author information is available at the end of the article

Abstract

The latest research on smart city technologies mainly focuses on utilizing cities' resources to improve the quality of the lives of citizens. Diverse kinds of control signals from massive systems and devices such as adaptive traffic light systems in smart cities can be collected and utilized. Unfortunately, it is difficult to collect a massive dataset of control signals as doing so in the real-world requires significant effort and time. This paper proposes a deep generative model which integrates a long short-term memory model with generative adversarial network (LSTM-GAN) to generate agent control signals based on the words extracted from newspaper articles to solve the problem of collecting massive signals. The discriminatory network in the LSTM-GAN takes continuous word embedding vectors as inputs generated by a pre-trained Word2Vec model. The agent control signals of sequential actions are simultaneously predicted by the LSTM-GAN in real time. Specifically, to collect the training data of smart city simulations, the LSTM-GAN is trained based on the Corpus of Contemporary American English (COCA) newspaper dataset, which contains 5,317,731 sentences, for a total of 93,626,203 word tokens, from written texts. To verify the proposed method, agent control signals were generated and validated. In the training of the LSTM-GAN, the accuracy of the discriminator converged to 50%. In addition, the losses of the discriminator and the generator converged from 4527.04 and 4527.94 to 2.97 and 1.87, respectively.

Keywords: Control signal, Simulation, Smart city, Word2Vec, LSTM-GAN

Introduction

Information and communication technologies (ICTs) play an important role in the development of smart and sustainable cities, which is an encompassing framework that includes not only physical infrastructure, but also human and social factors [1]. Smart cities are one of the main research topics based on Internet of Things (IoT) technology [2, 3]. In particular, the applications of smart cities require various integrated algorithms [4]. The diverse resources of smart cities are analyzed and utilized through technologies such as IoT, big data, social networks, and cloud computing, which improve the quality of the lives of citizens [5]. The development of smart cities currently involves the design and implementation of transportation, energy, traffic control, security, and other areas. The cost of physically installing these systems is very high in terms of both money and resources. However, controlling agents by smart control signals in real time instead of controlling the systems directly can significantly reduce these costs.

Concurrently, it is difficult to evaluate the performance of the models and technologies designed for smart cities [6]. For example, traffic control is one of the important topics covered in smart city research. Evaluating the assorted models and technologies is difficult when different cities implement different intelligent traffic controls [7]. Thus, simulation environments are constructed to validate the models and technologies [8]. In simulation environments, human behaviors and movements need to be considered. For example, very high costs will be incurred in a testing system that is utilized for detecting and tracking pedestrian movements. The simulation techniques aid the design of such a system while drastically reducing the costs; however, there is a disadvantage in that control signals must be collected from various agents to build a simulation environment. Although the simulation environment reduces the cost of collecting data in the experiment, in order to establish a simulation environment, some basic data is needed to formulate rules for the agent.

This paper proposes a deep generative model that generates agent control signals automatically by utilizing sentences extracted from news articles. The agent control signal includes the time, action, and place for a simulated agent in a smart city. Inspired by the great success of generative adversarial network (GAN) models in the areas of computer vision and natural language processing, GANs are utilized for text generation [9]. Because recurrent neural networks are more suitable than other deep neural networks when using sequential text data [10, 11], the GANs can be implemented using a recurrent neural network long short-term memory (LSTM) based model to generate the agent control signals based on text.

The proposed method consists of a newspaper article preprocessing phase, an LSTM-based GAN (LSTM-GAN) training phase, and an agent control signal generation phase. In the newspaper article preprocessing phase, the movement related words of an agent are extracted from the text dataset of selected newspaper articles. Specifically, movement related words are utilized to train a Word2Vec model, which generates the vector representations that encode patterns in the context of the extracted movement related words [12, 13]. In the LSTM-GAN training phase, the new architecture of the LSTM-GAN is designed, which stacks Word2Vec, GAN, and LSTM together for better generation of sentences from movement related words. In the agent control signal generation phase, the agent control signals are generated based on the trained LSTM-GAN and agent control signal maker. The LSTM-GAN consists of a deep LSTM-based generator network and a deep LSTM-based discriminator network. In particular, the discriminator of the LSTM-GAN utilizes the Bidirectional LSTM (Bi-LSTM) neural network to accommodate the control signals in both directions. The contributions of this paper are as follows:

- *Agent control signal generation based on natural language.* The movement related words, such as action (i.e., verb), place, and time expressions, are extracted from the newspaper articles in the COCA utilizing the part of speech tagging technology in natural language processing [14].

- *Expandability of the Word2Vec model.* Word2Vec is an embedding model used to generate word vectors that map each word to a continuous vector and represent the relationship between words. The Word2Vec model is trained to capture the syntactic structures of words and transform words into continuous vectors that are taken as input together with noise vectors to train the discriminator network in the LSTM-GAN. By using Word2Vec for preprocessing to generate agent control signals in the smart city simulation, we demonstrate the extensibility of utilizing Word2Vec in smart city simulations.
- *Control signal generation based on the LSTM-GAN.* A large-scale text dataset with 100 test sentences is built to ensure that the proposed framework can generate agent control signals for smart city simulations based on sentence generation.

The remainder of this paper is organized as follows: "Related work" section reviews the literature on the models pertaining to the present research; "Agent control signal generation from articles" section introduces the proposed method; "Experiment" section demonstrates the results of the experiments; "Discussion" section discusses the results of the experiments; and finally, "Conclusion" section concludes the paper.

Related work

Smart city simulation

Several previous studies on smart cities have conducted interdisciplinary research on the development of virtual cities with transportation, energy [15, 16], etc. The subject of smart cities still faces some challenges in its implementation, but more research projects on implementing smart cities have been conducted. Meanwhile, a photorealistic three-dimensional city simulation method has been recently proposed for training autonomous vehicles in a smart city simulation. Deep learning techniques such as convolutional neural networks (CNNs) have also been diversely applied to simulate the real-world behaviors of various agents and traffic systems [17].

In future smart cities, new information and communication technologies will manage urban resources better. The smart grid infrastructure is evolving into a complex system. Such a system can monitor and control the generation and consumption of energy in the power grid to improve energy efficiency. In [18], it implements and performs location extraction, location recognition, and location prediction on a position-based services (PBS) intermediate server by utilizing machine learning techniques; however, the user's position from the electronic device in the city is obtained in real time and the movement paths are predicted. Dynamic simulations of smart cities to test new resource optimization methods are introduced in [19]. A simulator based on software agents is built to create the dynamic behaviors of smart cities. It also simulates discrete heterogeneous devices that generate and consume energy. Thus, smart cities utilize multiple technologies to improve the implementation of transportation, energy, and traffic system services, which leads to higher levels of comfort for their citizens. In simulating smart city services, a technology that has significant potential is big data analysis. The most important part of the simulation system for training autonomous agents is generating various real-world situations based on the data acquired from the real-world; however, there are still many challenges in collecting the data of real-world events that occur to the real-world. Therefore, we propose a method for generating text data in this study, which is utilized to simulate the real behaviors of people.

Word embedding method

Natural language processing commonly includes two steps: data preprocessing and feature extraction. The data preprocessing comprises tokenization, cleaning, and lemmatizing phases [20]. The tokenization phase breaks sentences into words. The cleaning phase removes stop or function/non-content words to reduce the occupied database space and processing time. The lemmatizing phase groups the different inflected forms of a word together; thus, words are analyzed as a single item. Therefore, it collapses words with similar meanings into one representative word.

A natural language corpus consists of a large number of words, such that it requires significant computing resources to process. One-hot encoding is an encoding method that represents sentences or words as vectors with one and zero [21]; however, it leads to the "curse of dimensionality" by creating a new dimension for each word. Term Frequency–Inverse Document Frequency (TF-IDF) is another encoding method that represents how important a specific word is in a given document [22]. TF measures the total number of times that a given word appears in the document against the total number of words in the document, whereas IDF measures the proportion of a given word in the entire document. Similarly, word embedding representation is a language modeling technique utilized to map words to vectors of real numbers and learn distributed representations of words. It represents words in dense vector spaces with several dimensions. The important aspect of word embedding is that words occurring in similar contexts tend to be closer to each other in vector space; however, neither one-hot encoding nor TF-IDF elaborates on the semantics of the words. To avoid this drawback, the Word2Vec model consists of two architectures with Continuous Bag of Words (CBOW) and skip-gram for generating word embedding vectors. The CBOW model predicts the current word, given surrounding contextual words within a specific window, whereas the skip-gram model predicts the surrounding contextual words within a specific window given a current word [23]. In this study, the skip-gram architecture of the Word2Vec model is employed to predict continuous embedding vectors based on words extracted from sentences in the dataset. In fact, the Word2Vec model embeds the control signals extracted from newspaper articles.

Recurrent neural network (RNN)

The most popular approach to sentence generation is utilizing the RNN model. Currently, the most successful RNN model is the LSTM, where the gates in each neuron help the model predict the next word in a sentence based on the surrounding contextual words [24]. Many more models based on the LSTM network have been recently proposed, e.g., bidirectional LSTM [25]. Various studies have employed the LSTM for sentence generation, either directly [26] or as an embedded model [27]. In this paper, the LSTM is applied as a word predictor based on the input which is a sequence of words.

Generative adversarial network (GAN)

Recently, proposed by Goodfellow et al. in 2014, GAN was applied to various applications such as computer vision and natural language processing. GAN was proposed as a way of efficiently training deep generative neural networks. The fully connected GAN consists of two fully connected neural networks, a generator and a discriminator, where the generator attempts to produce realistic samples that fool the discriminator, while the discriminator attempts to distinguish real samples from generated ones [28]. Laplacian Pyramid GAN (LAPGAN) is a generative adversarial model utilizing the CNN within a LAP framework to generate images in a coarse-to-fine fashion [29]. Similarly, the GAN is a generative adversarial model utilizing input in the original GAN, which is applied to the class of an image, the attributes of an object, or embedded text descriptions of the image [30]. There are some obstacles in applying GAN to NLP. For example, one such obstacle is the discrete space of words that cannot be differentiated in mathematics. To this end, the sequence GAN (seqGAN) executes text generation by utilizing the softmax function over continuous values for word selections [27]. Inspired by seqGAN, the continuous RNN GAN (C-RNN-GAN) was proposed by Mogren; it works on continuous sequential data without the obstacles and is configured based on the Bi-LSTM to generate music melodies [31]. In addition, an enhanced GAN model that adds one more CNN discriminator based on the C-RNN-GAN to generate music melodies [32] has been proposed. In this study, a GAN model composed of an LSTM-based generator and a Bi-LSTM-based discriminator is trained upon the sequential information of text. The next section will detail how the proposed method automatically generates the agent control signal via the GAN model.

Agent control signal generation from articles

The proposed LSTM-GAN model to generate agent control signals from newspaper articles is depicted in Fig. 1. It consists of three major phases: newspaper article preprocessing, LSTM-GAN training, and agent control signal generation.

First, in the newspaper article preprocessing phase, the article extractor extracts newspaper articles from the database of the text corpus. The preprocessor, together with the additional place and action lists, transforms the extracted newspaper articles into the control signal embedding vectors via the pre-trained Word2Vec model. Second, in the LSTM-GAN training phase, the generator trainer utilizes the noise vectors as inputs to generate the fake control signals. The noise vector is a random distribution that is usually taken as input in the generator network. To map the noise vector to the new control signal vector, the generator network needs to ensure that the dimensions of the noise vector are the same as the dimensions of the embedded control signal. The discriminator is trained to distinguish the generated control signals from the embedded control signals. Finally, in the agent control signal generation phase, a sequence of the generated control signals concatenated with noise vectors is taken as input for the generator executor to generate control signals, which are converted into agent control signals via the agent control signal generator.

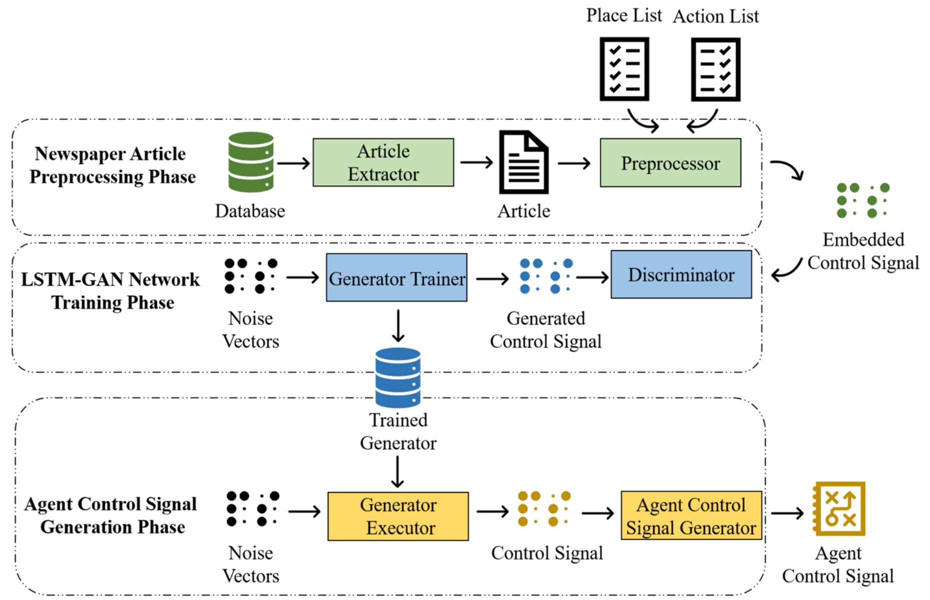

Fig. 1 Newspaper article-based agent control signal generation process

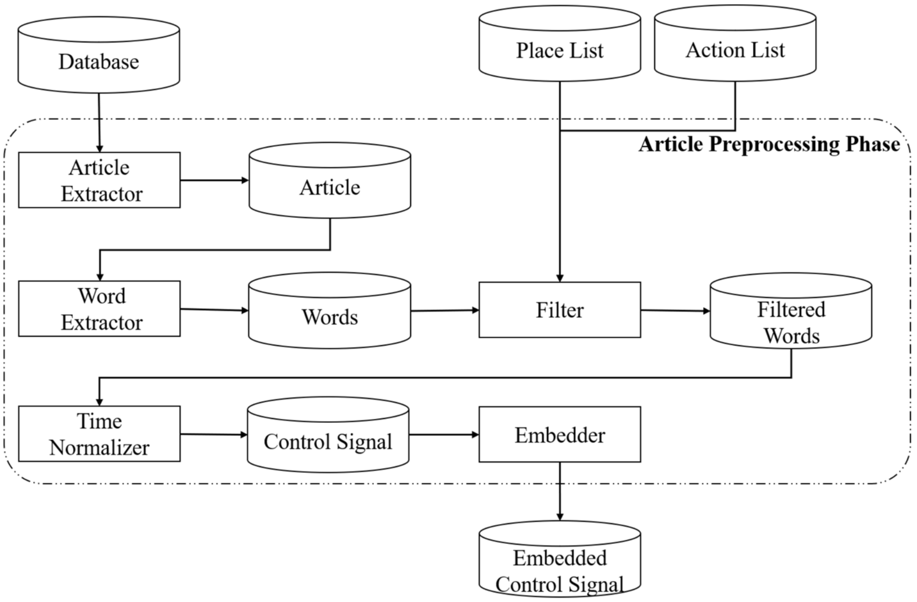

Fig. 2 Newspaper article preprocessing phase

Newspaper article preprocessing phase

The control signals are defined as a sequence of words denoting time, action, and place. The agent control signals are a set of control signals arranged according to the times that they are associated with. In Fig. 2, the database comprises a large number of newspaper articles, with the index number marked at the beginning of each article. The paragraphs in the newspaper articles are divided by the symbol "<p>". The article extractor first extracts the newspaper articles by recognizing the article index number from the database. The word extractor then extracts movement related words of time, action (i.e., verb), and place expressions from the sentences in a paragraph that are split by the symbol "<eos>". The movement related words are extracted from a single sentence to define the agent's action at a given time and place by utilizing Part-Of-Speech (POS) tagging and named-entity recognition (NER) in NLP. The movement related words extracted from sentences will be filtered based on a predefined action (verb) list and place list. Meanwhile, there are two ways of telling the time. The 12-h clock runs from 1 a.m. to 12 o'clock noon and then from 1 pm to 12 o'clock midnight. The 24-h clock uses the numbers 00:00 to 23:59 (midnight is 00:00). The time normalizer normalizes the filtered time words to the 24-h clock and outputs control signals. For example, to avoid any confusion when referring to '(in) the morning' from extracted words, the time word is normalized to the 24-h clock instead. The generated control signals are transformed into continuous control signal embedding vectors via the Word2Vec model.

LSTM-GAN network training phase

Figure 3 shows the deep training process of the LSTM-GAN network for generating new control signals based on embedded control signals from the newspaper articles. The LSTM-GAN network consists of a generator and a discriminator that are trained simultaneously with adversarial objectives. The generator has two LSTM layers, and the discriminator utilizes the Bi-LSTM model. The noise vectors are fed into the generator as the input, which represents a Gaussian distribution, and the control signals are generated by the generator. The generator takes the uniform noise vectors as inputs and generates control signals. The discriminator takes the generated control signals and the embedding control signal vectors as inputs. Furthermore, it distinguishes the control signals generated by the generator from the real ones in an adversarial manner.

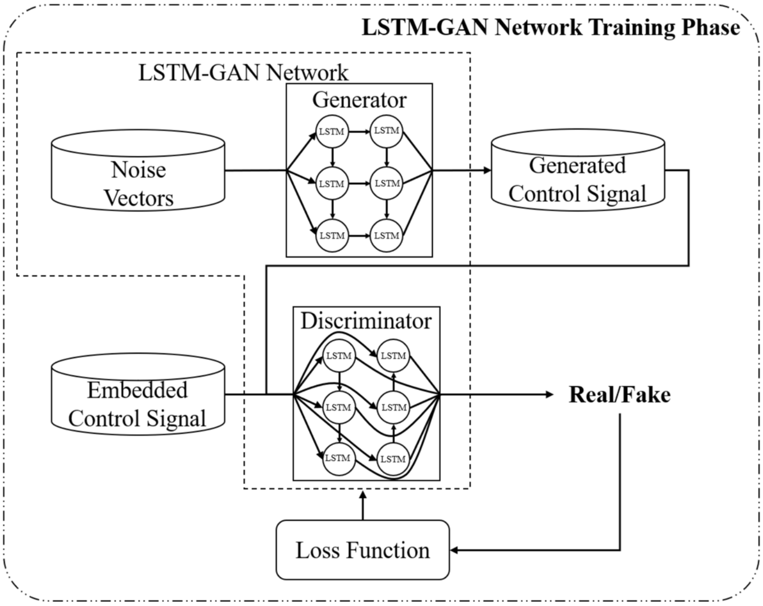

Fig. 3 LSTM-GAN network training phase

Equations (1) and (2) are employed as the loss functions, which are implemented jointly to train the discriminator and the generator. Equation (1) represents the loss function of the discriminator, and Eq. (2) represents the loss function of the generator, where x and z represent the real control signal embedding vector and the Gaussian normal noise vectors, respectively, G(z) denotes the fake control signal generated by the generator based on the noise vector, and finally, D(x) and D(G(z)) refer to the discriminative results of the real control signal and the fake control signal, respectively.

$$\min_{D} V(D, G) = -E_{x \sim p_{data}(x)}[log(D(x))] - E_{z \sim p_z(z)}[log(1 - D(G(z)))] \tag{1}$$

$$\min_{G} V(D, G) = -E_{z \sim p_z(z)}[log(D(G(z)))] \tag{2}$$

Agent control signal generation phase

Figure 4 shows the process of generating agent control signals. Various control signals are generated from the trained LSTM-GAN. Similarly, agent control signals are generated by the agent control signal maker based on the sorted control signals. Using noise vectors, the trained generator outputs control signals. The time feature for a particular control signal is defined as the start time and end time for that particular control signal. The probability calculator uses the start time together with the generated control

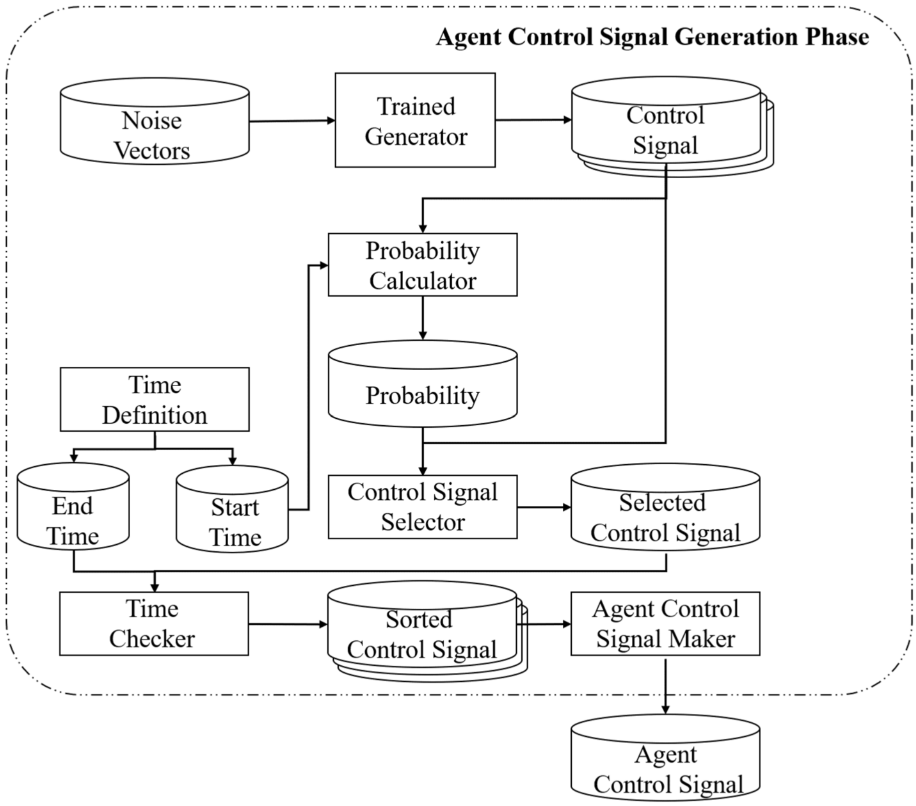

Fig. 4 Agent control signal generation phase

signal as input, then calculates the probability of the action occurring at the corresponding time according to the start time. To randomly select the control signal based on the probability, the probability and the control signal are fed into the control signal selector. The time checker takes the end time and the selected control signal as inputs and subsequently returns the control signal until the time of the control signal does not exceed the given time. Finally, the sorted control signal is generated. In the next section, a series of experiments are presented to prove the performance of the proposed system.

Experiment

In this section, the experimental setup and results are introduced to demonstrate the feasibility of the proposed LSTM-GAN model described in "Agent control signal generation from articles" section.

Data set and experimental environment

The COCA [12] is the largest American English corpus covering a variety of genres; it contains more than 500 million words. As shown in Table 1, experimental data were extracted from newspaper articles dated from 1990 to 2012 in this study.

Table 2 presents some of the hyperparameters that were applied during the implementation of the LSTM-GAN model. *Input* refers to the dimension of the input vector; *Output* refers to the dimension of the output vector; *Layer number* is the number of network layers; *Layer size* is the dimension of the network layer; *Learning rate* is a tuning parameter in the optimization algorithm that determines the step size at each iteration while approaching the minimum of a loss function; *Sequence length* denotes the length of the control signal; *Batch size* denotes the number of training words fed into the input layer in each iteration; and *Epoch* denotes the number of times that training is performed.

The experimental environment comprised Windows 10, i5-6400, NVIDA GeForce GTX 1050 2 GB, and DDR4 8 GB. The proposed LSTM-GAN network was implemented in Python using TensorFlow and the Natural Language Toolkit (NLTK) and Spacy libraries.

Table 1 Samples of newspaper articles in COCA

Articles of news value parameter (GAN)	
##3000001	<p> He is trying to make the best of it, but the days have seemed to pass like months for Lou Piniella … …. This is a stable situation, and if I do a job I 'll be around for a long time. "
##3000002	<p> Another North American indoor track and field Grand Prix season closes tonight with the USA … … Diane Dixon will be attempting to win her ninth title in the women 's 400
##3000003	<p> A century ago, this northern corner of the Florida Everglades was snake-infested swampland, … …Otherwise, the bulldozers @ @ @ @ @ @ @ @ @ @ the earth from which it rose
……	……
##4115379	<p> As Spain keeps the world economy on edge while it scrambles to recapitalize its once-mighty … … We deserve what we are getting. "<p>

Table 2 Parameters employed in the experiment

Parameter	Word2Vec	Generator (GAN)	Discriminator
Model	Skip-gram	LSTM	Bi-LSTM
Input	188	300	300
Output	300	300	1
Layer_number	1	2	2
Layer_size	300	100	100
Learning_rate	0.0001	0.1	0.1
Sequence_length	1	3	3
Batch_size	10,000	32	64
Epoch	10	16	16

Table 3 Processing of sentences

Paragraph			
<p> The one-mile turf course is an almost fluorescent green, lush from sunshine and a season of rest. The surrounding mile-and-an-eighth dirt track has long held its own magic, the kind that horses and horsemen love. Even when it is very quick, it is still forgiving. Nowhere else can a horse be galloping so close to you in the morning without your even hearing it. <p>			
Index	**Sentence**		
1	The one-mile turf course is an almost fluorescent green, lush from sunshine and a season of rest <eos>		
2	The surrounding mile-and-an-eighth dirt track has long held its own magic, the kind that horses and horsemen love <eos>		
3	Even when it is very quick, it is still forgiving <eos>		
4	Nowhere else can a horse be galloping so close to you in the morning without your even hearing it <eos>		
Index	**Time**	**Action**	**Place**
Word extraction			
1	–	Rest	–
2	–	Love	Track
3	–	Forgiving	–
4	Morning	Galloping, hearing	–
Information completion			
1	–	Rest	–
2	–	Love	Track
3	–	Forgiving	Track
4	Morning	Galloping, hearing	Track
Insufficient information control signal remove			
4	Morning	Galloping, hearing	Track
Word filtering			
4	Morning	Hear	Track
Time normalization			
4-1	08	Hear	Track
4-2	09	Hear	Track
4-3	10	Hear	Track
4-4	11	Hear	Track

Results of newspaper article preprocessing

Table 3 shows that the sentences, which were in one of the paragraphs in the newspaper articles, were split and indexed by the sent tokenize function in the NLTK library. The words contained in time, action, and place expressions were extracted from each sentence using the Spacy library.

The dataset used in this study did not have a sufficient number of training words denoting time, actions, and places. In general, if time and place expressions were included in a sentence in a paragraph, the same expressions were to be omitted if found in subsequent sentence(s). Thus, when time and place expressions were extracted, time and place related words in the preceding sentence(s) were to be adopted in the subsequent sentence(s). Recall that the control signal included three elements: time, action, and place expressions. As shown in Table 3, to ensure the applicability of the action and place related words in the simulation experiment, a predefined list of place expressions and high frequency action words was used.

The sentences without such expressions were eliminated. Furthermore, unclear time expressions in the extracted sentences were normalized. For example, although a time expression such as the word "morning" is a range representing word, it was set as a discrete value from 8:00 to 11:00. Similarly, each discrete time was assigned the same action and place. Through the newspaper article preprocessing method elucidated above, the control signals were processed into the training data, and the predefined list of high frequency action and place expressions shown in Table 4 was utilized. Each list contained 82 words. Only the words that appeared in a predefined list were considered as part of the training data.

The total number of final control signals extracted from the newspaper articles was 325,038. Table 5 presents some samples of the preprocessed control signals. Here, "control signals" refers to complete signals that include triplets comprising time, action, and location expressions. Animation is the simulation of movements by successively displaying a series of images. It is used in the smart city simulation to create the effect of traffic congestion, for example, to control the traffic arising from cars. Therefore, it is a behavior performed by an agent in a smart city simulation. The control signals served as agent control signals when they were matched with animations in simulation environments.

The 325,038 preprocessed control signals were input into the Word2Vec model to transform them into word vectors. Thus, 24 h, 82 actions, and 82 locations were embedded, as shown in Table 6.

Table 4 Predefined list of high frequency action and place

Index	High frequency action	Place
1	Host	Market
2	Stand	Farms
3	Pray	Bistro
4	Sound	Rink
…	…	…
82	Chat	Theater

Table 5 Word extracted from sentences

Time	Action	Place
20	Server	Restaurant
24	Server	Restaurant
20	Enter	Hotel
24	Enter	Hotel
02	Keep	Bed
08	Say	Museum
11	Say	Museum
12	Drive	Road
……	……	……
19	Drive	Road

Table 6 Embedded words

Word	Vector
01	0.18026, 0.06278, – 0.08361, ……, 0.01421, 0.11844, 0.22934
02	0.21194, 0.07123, – 0.09841, ……, 0.01803, 0.13853, 0.26796
…	…
24	0.23031, 0.07799, – 0.10717, ……, 0.01618, 0.14913, 0.29573
Spend	0.06482, 0.02008, – 0.02964, ……, 0.00745, 0.04541, 0.08771
Find	0.11458,0.04738, – 0.05540, ……, 0.00716, 0.07596, 0.14747
…	…
Chat	0.06944, 0.02238, – 0.03007, ……, 0.00636, 0.04652, 0.08742
Restaurant	0.13838, 0.05084, – 0.06026, ……, 0.00149, 0.08690, 0.17570
Beach	0.13931, 0.04394, – 0.06520, ……, 0.01303, 0.09113, 0.17620
……	……
School	0.17103, 0.05772, – 0.07892, ……, 0.02009, 0.10672, 0.20040

Training results of LSTM-GAN network

Figure 5a shows the accuracy of the discriminator during the training process. The blue line in the figure represents the *Real accuracy*, which indicates the accuracy of the discriminator in determining real samples as being real. The orange line denotes the *Fake accuracy*, which represents the accuracy of the discriminator in determining the generated fake samples as being fake. The gray line denotes the accuracy of the discriminator calculated using both the *Real accuracy* and the *Fake accuracy*; it represents the accuracy of the discriminator for all the samples [33]. The *Real accuracy* rate was 50% at Epoch 1, and it converged to 90% after Epoch 6. The *Fake accuracy* rate was 50% at Epoch 1, and it converged to 11% after Epoch 6. The *Accuracy* fluctuated substantially at Epoch 5 and converged to 50% after Epoch 6. At the beginning of the training, because the discriminator had not been trained, the accuracy rates for both real and fake samples were 50%. The increase of the *Real accuracy* indicates that the discriminator gradually improved its performance through training, and the decrease of the *Fake accuracy* indicates that the fake samples generated by the generator were similar to the real samples owing to training. The *Accuracy* converged to 50% for all the samples, which means that the discriminator and generator were well trained [34]. Figure 5b shows the training loss of the

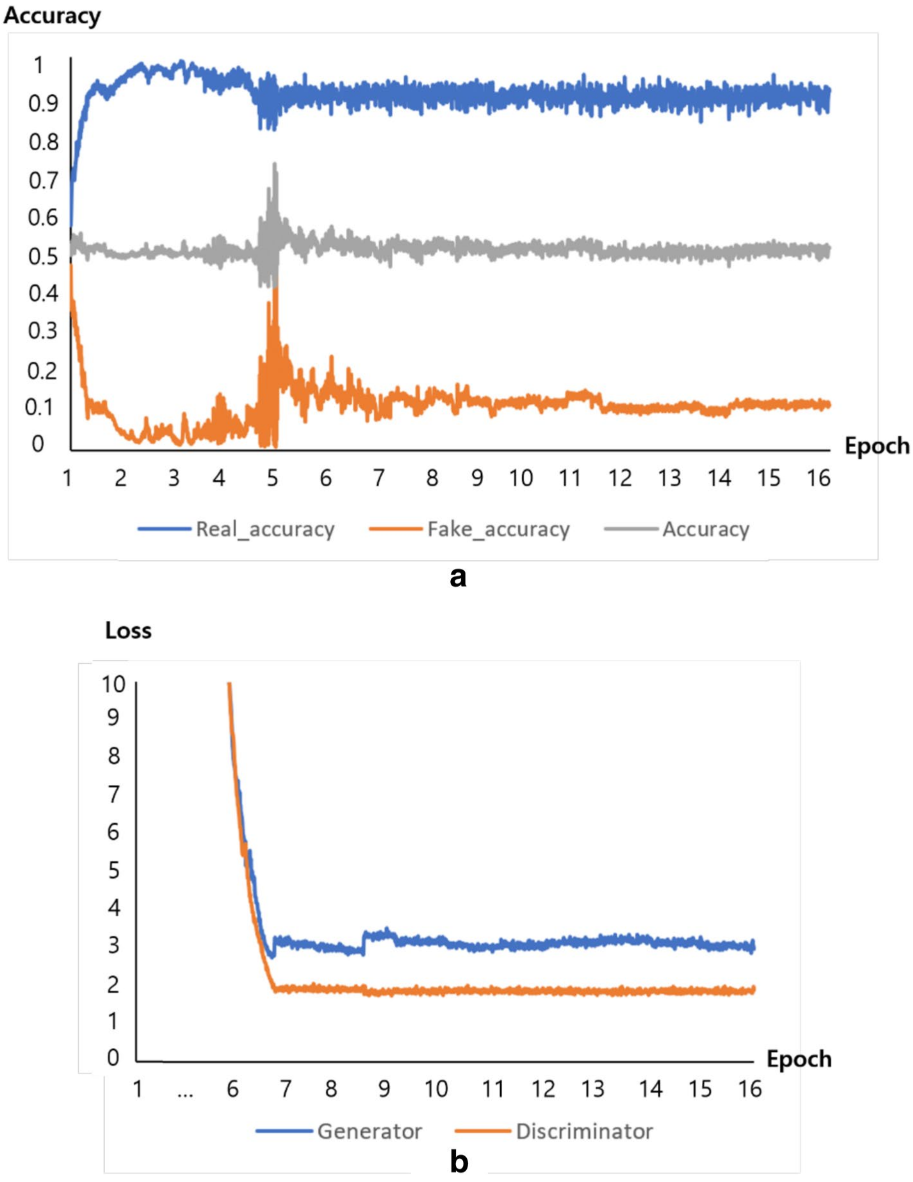

Fig. 5 Training results of LSTM-GAN

LSTM-GAN. The loss of the generator network yielded 4527.04 in Epoch 7 and converged to 2.97 in Epoch 15. The loss of the discriminator network was 4527.94 in Epoch 7 and converged to 1.87 in Epoch 15. In addition, when the loss of the generator and that of the discriminator converged, the loss value of the discriminator was lower than that of the generator. This was because the discriminator could give feedback using the actual data while the generator learned through the feedback produced by the discriminator.

Results of agent control signal generation

Table 7 shows the control signals generated by the trained LSTM-GAN. The generated control signal comprises time, action, and place expressions.

Table 7 Control signal generated using the trained generator

Time	Action	Place
14	Spend	Bar
19	Find	Valley
04	Know	Airport
04	Talk	College
14	Propose	Hospital
22	Lock	Building
21	Work	University
10	Read	Backyard
16	Arrive	Bank
12	Mix	Cafe
11	Walk	Road
……	……	……
12	Sleep	Beach

Table 8 Results of probability calculation

Time	Action	Probability
21	Know	0.011
	Imagine	0.041
	Tell	0.020
	Think	0.015
	…	…
	Stay	0.027
22	Know	0.011
	Come	0.012
	Tell	0.041
	Hear	0.031
	…	…
	Stay	0.027
23	Spend	0.010
	Know	0.050
	Talk	0.035
	Come	0.033
	…	…
	Sleep	0.025
……	……	……
20	Spend	0.010
	Know	0.035
	Come	0.012
	Sleep	0.041
	……	……
	Hear	0.031

The probabilities of time-dependent actions were computed as shown in Table 8 to generate the agent control signals. Generally, the probabilities of time-dependent actions were determined based on the frequency of each action in the control signals generated from the trained LSTM-GAN. However, in this study, to generate more diverse agent control signals, actions were not extracted based on the highest probability at a given time; instead, they were randomly generated based on whether the probability was high or low. Thus, the actions with low probabilities were occasionally extracted.

In the agent control signal generation phase, agent control signals were generated from 21:00 to 20:00 on the next day, as shown in Table 9. The start time was defined as 21:00, and control signals comprising time, action, and place expressions were generated every hour until the end time of 20:00 the following day. Therefore, after the start time, a 24-h agent control signal will be generated for one day. The results obtained in the experiments will be discussed and analyzed in the next section.

Discussion

In this section, we enumerate some advantages of combining the LSTM and GAN models to generate control signals automatically. Concurrently, the performance of the LSTM-GAN is discussed by comparing the text-based GAN-generated control signals with the ground truth control signals available in datasets.

It is evident that GANs have achieved substantial success in computer vision with regard to generating hyper-realistic images. Building on this success, image GANs have recently been extended to tasks such as data augmentation. In this paper, to generate text-based control signals for the simulation, we adopted an LSTM-based generator as the generator network in a GAN; however, GANs have posed the following problems when applied to text generation. First, when an LSTM-based generator is employed as the generator network in a GAN, the latent noise vector is the

Table 9 Agent control signal

Agent	Time	Action	Place
Agent control signal 1	21	Visit	Pool
	22	Wake	Rink
	23	Find	Harbor
	…	…	…
	20	Stay	Theater
Agent control signal 2	21	Work	University
	22	Wake	Rink
	23	Chat	Lawn
	…	…	…
	20	Hear	Camp
Agent control signal 3	21	Stay	Farm
	22	Stand	Sea
	23	Say	Bank
	…	…	…
	20	Complain	College
…	…	…	…
Agent control signal N	21	Visit	School
	22	Go	Hall
	23	Want	School
	…		
	20	Go	House

input hidden state of the LSTM, and the output of the generator is the output sentence yielded by the LSTM. In this paper, instead of training the LSTM to minimize cross-entropy loss with respect to target one-hot vectors, we trained it to increase the probability of the discriminator network classifying the control signals as "real." While decoding with an LSTM at every time step, we chose the next word by picking the word with the maximum probability from the output of a softmax function. This "picking" operation is non-differentiable. We consider this to be crucial because to train the generator to minimize the term 1−D(G(z)) in the loss function, we need to feed the output of the generator into the discriminator and backpropagate the corresponding loss of the discriminator. For these gradients to reach the generator, they have to go through the non-differentiable "picking" operation at the output of the generator. This is problematic, as backpropagation relies on the differentiability of all the layers in the network. In contrast, this is perfectly feasible when the generated data is continuous, such as image data. In recent times, various methods have been proposed to circumvent this problem. seqGANs are text generation GANs that employ reinforcement learning; however, reinforcement learning-based methods are known to generally yield very poor sentence quality owing to high variance gradient estimates. RelGAN stands for relational generative adversarial networks for text generation [35]; it is based on using Gumbel-softmax for a continuous approximation of the softmax function to effectively model long-term dependencies in the text. AGNL stands for adversarial generation of natural language [36], which is based on the continuous output of the generator. In contrast to the aforementioned methods, the LSTM-GAN model proposed in this study eliminated discrete spaces altogether by employing the continuous output of the generator. Specifically, it was designed to generate text-based control signals; recall that this model integrates word embedding language models such as LSTM and GANs without adopting reinforcement learning, as shown in Fig. 1. Recall that in the first phase, namely, the newspaper article preprocessing phase, newspaper articles were extracted from the database of the text corpus, which were then transformed into control signal embedding continuous vectors via the pre-trained Word2Vec model. Thereafter, in the second phase, that is, the LSTM-GAN network training phase, the noise vectors were used as inputs to generate fake control signals. The discriminator of the proposed LSTM-GAN was then trained to distinguish the generated control signals from the embedded control signals.

Furthermore, text-based agent control signals need to be applied to the subsequent simulation experiment for them to be matched with the animations in the simulation experiment. To solve this problem, the agent control signals were derived from sentences extracted from actual newspaper articles that reflect actual people performing specific actions at specific places. That is, the control signals generated by the LSTM-GAN have a format that comprises time, action, and location expressions in a sequential order. They have a well-arranged structure when compared with the control signals extracted from the newspaper articles. For example, "21, work, university" represents working at the university at nine in the evening. Therefore, the agent control signals generated based on the proposed method were utilized in the simulation experiment, which increased the training efficiency substantially.

Conclusion

This paper proposed a method for generating agent control signals based on words extracted from newspaper articles. First, to obtain meaningful words from newspaper articles, expressions related to time, actions, and places were extracted using the POS Tagging and NER technologies. Second, the control signals for simulation experiments were selected by filtering the action, place, and time normalization. Third, the extracted control signals were embedded using the Word2Vec model to express the relationships between the words in the signals, and the embedded control signals were in turn applied to deep learning models. Fourth, the LSTM-GAN was trained using the embedded control signals to generate new control signals. Fifth, based on the control signals generated by the LSTM-GAN and the defined start time, the probabilities of actions occurring at various times were calculated. Based on the calculated probabilities, the control signals were selected. Sixth, the selected control signals were sorted based on the defined start and end times; thereafter, the agent control signals were generated. Furthermore, to validate the proposed method, an agent control signal generation experiment was performed. In the training of the LSTM-GAN, the accuracy of the discriminator converged to 50%. In addition, the losses of the discriminator and the generator converged from 4527.04 and 4527.94 to 2.97 and 1.87, respectively.

The agent control signals generated using the proposed method needed to be applied to the subsequent simulation experiment in order for them to match the animations in the simulation experiment. Because the agent control signals were derived from sentences extracted from actual newspaper articles, they reflected specific actions performed by people at specific places. Thus, the agent control signals generated based on the proposed method were employed in the simulation experiment, which increased the training efficiency.

This study is, to the best of our knowledge, the first one to utilize an LSTM-GAN model integrating Word2Vec to generate large scale agent control signals based on sentences that include words denoting times, actions, and locations for smart city simulation tasks. As emphasized above, the generation of text-based artificial agent control signals can be extremely useful, particularly with regard to imbalanced datasets and data augmentation for smart city simulations. The proposed agent control signals are a text-based dataset for real-time pedestrian simulation at non-specific places or places without addresses, which are common factors in general road traffic modeling. To ensure that the newly designed text-based agent control signals are able to handle the smart city traffic congestion simulation, we need to modify the proposed data structure for people and vehicle flow simulation at specific areas. To do this, it will be necessary to define people's movement in more detail, namely by integrating campus squares with specified addressed into the existing road networks for real-time pedestrian simulation. In the future, we need to build a text data–driven road traffic dataset for simulating pedestrian motion based on more articulated empirical data that keeps track of the pedestrian's position and includes motion capture recordings at location with specified addresses to generate more exact motion data.

Acknowledgements
This research was supported by The Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF—2017S1A5A2A01026286).

Authors' contributions

EK has proposed the main ideas and is the main contributor of this article. SJ and SL have contributed to doing our experiments. YS is the second contributor and the corresponding author of this article. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Department of Computer Science and Engineering, Shinhan University, Kyunggi 11644, Republic of Korea. [2] Department of Multimedia Engineering, Dongguk University-Seoul, Seoul 04620, Republic of Korea.

Received: 6 January 2020 Accepted: 19 October 2020
Published online: 03 November 2020

References

1. Bibri SE (2018) A foundational framework for smart sustainable city development: theoretical, disciplinary, and discursive dimensions and their synergies. Sustain Cities Soc 38:758–794
2. Park JH, Salim MM, Jo JH, Sicato JCS, Rathore S, Park JH (2019) CIoT-Net: a scalable cognitive IoT based smart city network architecture. Hum Comput Inf Sci 9(29):29–49
3. Lee Y, Rathore S, Park JH, Park JH (2020) A blockchain-based smart home gateway architecture for preventing data forgery. Hum Comput Inf Sci 10(9):1–14
4. Jeong YS, Park JH (2019) IoT and smart city technology: challenges, opportunities, and solutions. J Inf Process Syst 15(2):23–238
5. Del Esposte ADM, Santana EF, Kanashiro L, Costa FM, Braghetto KR, Lago N, Kon K (2019) Design and evaluation of a scalable smart city software platform with large-scale simulations. Fut Gener Comput Syst 93:427–441
6. Mallapuram S, Ngwum N, Yuan F, Lu C, Yu W (2017) Smart city: the state of the art, datasets, and evaluation platforms. In: 2017 IEEE/ACIS 16th international conference on computer and information science (ICIS), pp. 447--452. IEEE, Wu Han
7. Galán-García JL, Aguilera-Venegas G, Rodríguez-Cielos P (2014) An accelerated-time simulation for traffic flow in a smart city. J Comput Appl Math 270:55–563
8. Mesquita R, Campos-Rebelo R, Barros JP (2019) model based simulation for a smart city project based on LoRa. In: IECON 2019–45th annual conference of the IEEE industrial electronics society. IEEE, Lisboa, pp. 5868—5873
9. Li Y, Pan Q, Wang S, Yang T, Cambria E (2018) A generative model for category text generation. Inf Sci 450:301–315
10. Alayba AM, Palade V, England M, Iqbal R (2018) A combined CNN and LSTM model for arabic sentiment analysis. In: International cross-domain conference for machine learning and knowledge extraction, Springer, Hamburg, pp. 179--191.
11. Kwon DH, Kim JB, Heo JS, Kim CM, Han YH (2019) Time series classification of cryptocurrency price trend based on a recurrent LSTM neural network. J Inf Process Syst 15(3):694–706
12. Yilmaz S, Toklu S (2020) A deep learning analysis on question classification task using Word2vec representations. Neural Comput Appl 1:20
13. Yuan X, Wang S, Wan L, Zhang C (2019) SSF: sentence similar function based on Word2vector similar elements. J Inf Process Syst 15(6):1503–1516
14. https://www.english-corpora.org/coca/
15. Sangaiah AK, Medhane DV, Bian GB, Ghoneim A, Alrashoud M, Hossain MS. Energy-aware green adversary model for cyber physical security in industrial system. IEEE Trans Indu Inf. Doi: https://doi.org/10.1109/TII.2019.2953289.
16. Sangaiah, A. K., Hosseinabadi, A. A. R., Sadeghilalimi, M., Zhang, W.: Energy Consumption in Point-Coverage Wireless Sensor Networks via Bat Algorithm. IEEE. Access. Doi: https://doi.org/10.1109/ACCESS.2019.2952644.
17. Chu PM, Wen M, Park J, Kaisi H, Cho K (2019) Three-dimensional simulation for training autonomous vehicles in smart city environments. In: 2019 international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData), IEEE, Atlanta , pp. 84–853.
18. Sangaiah AK, Medhane DV, Han T, Hossain MS, Muhammad G (2019) Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics. IEEE Trans Industr Inf 15(7):418–4196
19. Pilehvar MS, Benzaquen J, Shadmand MB, Pahwa A, Mirafzal B, McDaniel J, Rogge D, Erickson J (2018) Modeling, control, and stability of smart loads toward grid of nanogrids for smart cities. In: IECON 2018—44th annual conference of the IEEE industrial electronics society, IEEE, Washington, DC , pp. 4045–4050.
20. Vijayarani S, Ilamathi MJ, Nithya M (2015) Preprocessing techniques for text mining-an overview. Int J Comput Sci Commun Netw 5(1):7–16
21. Rodríguez P, Bautista MA, Gonzalez J, Escalera S (2018) Beyond one-hot encoding: lower dimensional target embedding. Image Vis Comput 75:21–31
22. Zhang H, Xiao X, Mercaldo F, Ni S, Martinelli F, Sangaiah AK (2019) Classification of ransomware families with machine learning based on N-gram of opcodes. Fut Gener Comput Syst 90:211–221
23. Ma L, Zhang Y (2015) Using Word2Vec to process big text data. In: 2015 IEEE international conference on big data (Big Data), IEEE, Santa Clara, pp. 2895--2897
24. Kang J, Jang S, Li S, Jeong YS, Sung Y (2019) Long short-term memory-based malware classification method for information security. Comput Electr Eng 77:366–375

25. Liu G, Guo J (2019) Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 337:325–338
26. Pawade D, Sakhapara A, Jain M, Jain N, Gada K (2018) Story scrambler-automatic text generation using word level RNN-LSTM. IJITCS 10(6):44–53
27. Yu L, Zhang W, Wang J, Yu Y (2017) Seqgan: sequence generative adversarial nets with policy gradient. In: Thirty-first AAAI conference on artificial intelligence. pp. 2852–2858
28. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Advances neural information processing systems conference, pp. 2672–2680
29. Denton EL, Chintala S, Fergus R (2015) Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances Neural Information Processing Systems Conferences, pp. 1486–1494
30. Mirza M, Osindero S. Conditional generative adversarial nets. arXiv Preprint, arXiv:1411.1784
31. Mogren O. C-RNN-GAN: continuous recurrent neural networks with adversarial training. arXiv preprint arXiv:1611.09904
32. Li S, Jang S, Sung Y (2019) Automatic melody composition using enhanced GAN. Mathematics 7(10):883–896
33. Santhanam GK, Grnarova P. Defending against adversarial attacks by leveraging an entire GAN. arXiv preprint arXiv:1805.10652
34. Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA (2018) Generative adversarial networks: an overview. IEEE Signal Process Mag 35(1):53–65
35. Nie W, Narodytska N, Patel A (2018) Relgan: relational generative adversarial networks for text generation. In: International conference on learning representations, Vancouver, pp. 1–20
36. Rajeswar S, Subramanian S, Dutil F, Pal C, Courville A. Adversarial generation of natural language. arXiv preprint arXiv:1705.10929.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.