ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

PCON-2019

National Conference and Seminar on Innovations in Engineering & Technology

Govt Polytechnic College, Palakkad

Vol. 8, Special Issue 1, January 2019

Survey on Text Generation Based on Generative Adversarial Nets with Latent Variable

Remya K R[1], Jayasree M[2]

PG Student, Department of Computer Science and Engineering, Government Engineering College, Palakkad, Kerala, India[1]

Assistant Professor, Department of Computer Science and Engineering, Government Engineering College, Palakkad, Kerala, India[2]

Abstract: Text generation techniques can be applied for improving language models, machine translation, summarization, and captioning. One mainstream approach for text generation is by modeling sequence via recurrent neural network. In this paper, a survey is conducted on the recent text generation techniques based on GAN, SeqGAN, and RNN. Mainly conduct survey on a paper which has model utilizing Generative Adversarial Net (GAN) to create realistic text. Rather than utilizing standard GAN, consolidate Variational Auto Encoder (VAE) with generative adversarial net is used. The utilisation of high-level latent random variables is useful to learn the data distribution that takes care of the issue that generative adversarial net always emits the similar data. Here VGAN model has been used where the generative model is made out of recurrent neural network and VAE. The discriminative model is a convolutional neural network and it trains the model via policy gradient method. The VGAN is applied to the task of text generation and contrast it with other recent neural network based models, for example, recurrent neural network language model and SeqGAN. It evaluates the performance of the model by calculating negative log-likelihood and the BLEU score. Here experiments have been conducted on three benchmark datasets, and results demonstrate that a model beats different past models.

Keywords: Generative Adversarial Net, Variational Auto-encoder, VGAN, Text Generation

I. INTRODUCTION

Automatic text generation is a research area which has a lot in common. This area has the goal of creating a readable and coherent text for a specific user. In automatic text generation a computer automatically creates natural language, e.g. English, Chinese, or Greek, from a computational representation. Automatic text generation is the generation of natural language texts by computer which has vital role in natural language processing and artificial intelligence. For example, writing comments, weather reports and even poems can help us generate text. Machine translation, text summary, answering questions and dialog system are also important.

The main objective of the text generation is to model the language. Language modelling is one of the main problem in Natural Language Processing (NLP). Language models can be used at character, n-gram, phrase or even paragraph level. In the text generation, mapping is a step in which assign an arbitrary number to a character/word in the text. In this method, all unique characters/words are mapped to a number. This is important, because machines understand numbers far better than text, and this subsequently makes the training process easier. It must also be noted here that it have been used character level mappings and not word mappings. However, when compared with one another, a word-based model shows a lot of higher accuracy as compared to a character-based model. This is as a result of the latter model needs a far larger network to find out long-term dependencies because it not solely has to remember the sequences of words, however additionally should learn to predict a grammatically correct word. However, in case of a word-based model, the latter has already been taken care of.

Nowadays, it is available in audio, video, text, time series, sensor data and so on. One special thing about this type of data is that if two events occur in a certain time frame, the occurrence of event A before event B is a completely different scenario compared to that of event B. However, it is hardly important in conventional machine learning problems whether a particular data point has been recorded before the other. This consideration gives a different approach to sequence prediction problems. Text, a stream of characters lined up one after the other, is a difficult thing

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

PCON-2019

National Conference and Seminar on Innovations in Engineering & Technology

Govt Polytechnic College, Palakkad

Vol. 8, Special Issue 1, January 2019

to crack. This is because a model can be trained in the handling of text to make very precise predictions using the previous sequences, but a false prediction can make the whole sentence meaningless. However, in case of a problem with numerical sequence prediction, even if a prediction goes completely to the south, it could still be considered a valid prediction (perhaps with a high prediction). But it wouldn't hit the eye. This is what challenges text generation.

Text generation can widely used in many of the applications. Recurrent Neural Networks (RNNs) based text generation models have been widely used for generative tasks such as language modeling, machine translation, voice recognition, captioning of images. Language Modeling is that the task of predicting what word comes next. For an example, given the sentence I am writing a, the next word is to be letter, sentence, blog post, etc. In other words, given a sequence of words x(1), x(2),..,x(t), language models calculate the likelihood distribution of subsequent word x(t + 1).

There are various methodologies for text generation and they are Generating text with neural nets such as convolutional neural network, recurrent neural network and Long-Short Term Dependency. Generative adversarial nets (GAN)is always used to generate realistic text. This survey aims to discuss different ways of generating text using different neural networks and thus highlight the advantages and disadvantages of each. Such a comparative study is very important as there are wide range of applications in natural language processing and artificial intelligence using text generation techniques. This survey will provide some insights for choosing the right methodology to propose a text generation.

This paper is organized as follows: Section II gives a model description of the generative model, called VGAN, by combining Variational Auto - Encoder(VAE) and Generative Adversarial Nets(GAN) and adversarial training of VGAN. Section III discusses about various approaches used for text generation and their advantages and disadvantages. It provides a comparison between the methodologies. Section IV discusses the future scope of text generation techniques. Section V gives a brief conclusion of text generation with latent variables.

II. MODEL DESCRIPTION OF THE GENERATIVE MODEL OF VGAN

Text generation techniques can be used for improving language models, machine translation, summarization, and captioning. Generative models can work with multimodal outputs and can be prepared with missing data and can give expectations on missing data. Recurrent neural networks (RNNs) based models have been utilised generally for generative tasks such as language modeling, machine translation, speech recognition, and image captioning. Words from the previous time steps are input to the next time step iteratively in the generative RNN models. Reinforcement learning is always used [1] to optimize the model when GAN is applied to the task of text generation [2]. While GAN can generate realistic texts, even poems, there is an obvious disadvantage that GAN always emits similar data [3]. GAN usually uses recurrent neural networks as a generator for text generation. The recurring neural network mainly contains two parts: the transition from the state to the output and the mapping from the state to the output. This may not be enough to learn how highly structured data such as text are distributed. In oreder to learn generative sequence models, suggest that high latent random variables be used to model the observed variability. Combine recurrent neural networks with Variational Auto Encoders (VAE) [4] as G.

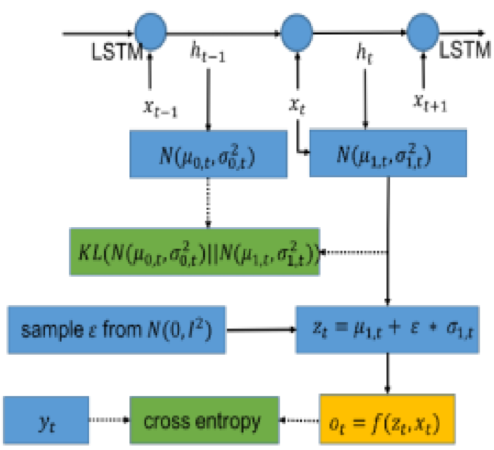

Fig. 1. The structure of the generator G_θ . The generator is composed of LSTM and VAE. x_t denotes the input at timestep t; h_t denotes the LSTM [14].

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

PCON-2019

National Conference and Seminar on Innovations in Engineering & Technology

Govt Polytechnic College, Palakkad

Vol. 8, Special Issue 1, January 2019

The generative model contains a VAE at every time step. The prior distribution of the standard VAE is usually a standard normal distribution. The current prior distribution depends on the hidden state h_{t-1} at the previous moment, unlike the standard VAE, and adding the hidden state as an input helps to reduce the long - term dependency of sequential data. The model is described in Fig. 1.

If the stochastic gradient descent algorithm is used to optimize the model directly, it is difficult to derive certain VAE parameters. To solve the problem, introduce the "reparametrization trick". Pre-train the generative model via SGVB before adversarial training. For example, given the input X_s = (S, i, like, it), and the target output is Y_s = (i, like, it, E), where S and E are respectively the start token and the end token of a sentence.

Here choose the convolutional neural network as the discriminative model, which has proved a great success in the text classification task [5] [6]. The training process of discriminator is showed in Fig. 2.

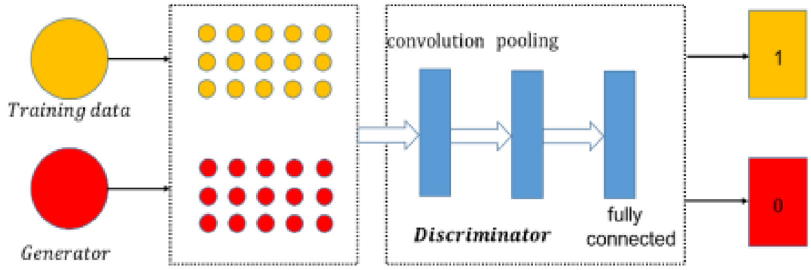

Fig. 2. The illustration of the discriminator. The discriminator D φ is trained by using the true training data and the fake data generated by generator [14].

III. DIFFERENT METHODOLOGIES FOR TEXT GENERATION TECHNIQUES

A. Recurrent Neural Network Based Language Model: Sequential data prediction is regarded by many as a key element in the learning of machines and artificial intelligence. The goal of statistical language modeling is to predict the next word in the context of textual data; thus, it deals with the problem of sequential data prediction when structuring language models. Language models for real-world speech recognition or machine translation systems are based on enormous amounts of data, and the popular belief is that it needs more data. Mikolov et al. [7] proposes and approach for trained at several times, in which all training corpus data are presented sequentially. Language modelling, it has decided to study recurrent neural networks for sequential data modeling. It has been used an architecture in this work, which is usually called a simple recurrent neural network or Elman network [8]. Networks are trained at several times, in which all training corpus data are presented sequentially.

The main advantage of this methodology is to predict the next word from the textual data given context and it is dealing with sequential data problem when language models to be constructed. This version of the recurrent neural network is probably the simplest possible and very easy to implement and train. The main issue of this methodology is recurrent neural networks do not use limited size of context. By using recurrent connections, information can be used arbitrarily long in these networks. It is also often argued, however, that learning long-term dependencies through stochastic gradient can be quite difficult.

B. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient Generating synthetic sequential data that imitates the real one is a major problem in unsupervised learning. In this paper [2] follow the sequence generation process and consider it as a sequential decision-making process. The generative model is treated as a reinforcing learning agent (RL); the state is the tokens generated so far and the action is the next token. In other words, this paper focuses on sequence discrete data generation. Here it has promising method has to be used, in which Generative Adversarial nets(GAN). Discriminator tries to correctly distinguish the true data and the fake model generated data. Generator attempts to generate high-quality data to fool discriminator. Ideally, when D cannot distinguish the true and generated data, G nicely fits the true underlying data distribution. Modelling Sequence Generation through SeqGAN via Policy Gradient and model architectures contain generator and discriminator. In the Discriminative Model for Sequences, deep discriminative models such as Deep Neural Network (DNN), Convolutional Neural Network (CNN) and recurrent neural network have demonstrated a high degree of performance in complex sequence classification tasks. In this paper, it has chosen the CNN as discriminator that was recently shown to be very effective in the text

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

PCON-2019

National Conference and Seminar on Innovations in Engineering & Technology

Govt Polytechnic College, Palakkad

Vol. 8, Special Issue 1, January 2019

classification [9] (token sequence). Most discriminative models can only be classified for a whole sequence instead of the unfinished one. In this paper, also focus on the situation in which the discriminator predicts that a finished sequence is likely to be real one. Similarly, in the Generative Model for Sequences, as a generative model, use Recurrent Neural Networks [10] (RNNs).

The main advantage of this methodology is [2]; SeqGAN has shown excellent performance in creative sequences for three real world scenarios, i.e. poems, language of speech and music generation. SeqGAN has demonstrated excellent performance in the creative sequences. The main limitation of this is the stability of SeqGAN depends on the training strategy and the discriminator cannot get fully trained and thus will provide a misleading signal gradually.

C. A Recurrent Latent Variable Model for Sequential Data: In this paper, [11] investigate the inclusion of latent random variables in the hidden state of a recurrent neural network (RNN) by combining the variation encoder elements. Neural network (RNN) by combining variational autoencoder elements. It argues that the variational RNN (VRNN) can model the type of variability observed in highly structured sequential data, such as natural speech, by using high - level latent random variables. Learning generative sequence models is a long – standing challenge in machine learning and is historically the field of dynamic Bayesian networks (DBNs) such as hidden Markov models (HMMs) and Kalman filters. Now the mainstream is recurrent neural network (RNN) is based on DBNs are simple (HMM state space are single set of mutually exclusive states), Training DBNs are hard (MCMC) and RNNs have a richly distributed internal state representation, RNNs have flexible non-linear transition functions, and RNNs can be trained by Back propagation. For latent variable model, use neural networks as the (generative) transformation from the latent space to the original feature space. For both training and inference, latent variable z must be inferred. This is a generative model in which inferring the latent variable is hard. In case of Variational recurrent Neural Network(VRNN), It is an application of the VAE at every time step and Latent variables can model noise in a structured way.

The VRNN model is evaluated for two tasks: (1) modeling natural speech directly from the raw audio waveforms; (2) modeling manual writing.

Speech Modelling: Train models to model raw audio signals directly, represented as a 200-dimensional frame sequence. Each frame corresponds to the actual amplitude of 200 raw acoustic samples in a row. Generation of handwriting allow each model to learn a sequence of (x, y) coordinates along with binary pen-up / pen-down indicators using the IAMOnDB data set consisting of 13,040 handwritten lines written by 500 authors [12] and prepare and split the dataset. Models of this survey compare the VRNN models with the standard RNN models with two different output functions: a simple Gaussian (Gauss) and a Gaussian (GMM) mixture. For each data set, it perform an additional set of experiments without the conditional prior (VRNN-I) for a VRNN model. Comparison between the VRNN model VGAN model with other RNN-based models including a VRNN model without introducing temporal dependencies between latent random variables.

The advantages of this methodology are more efficient use of previous experience, multiple learning. More efficient use of previous experience, multiple learning. This is important when it is costly to gain real world experience, you can make full use of it. The memory-learning updates are incremental and do not converge rapidly, so that multiple passes with the same data are beneficial, especially if there is little variation in immediate results. Coming to the limitation, in which it is harder to use multi - step learning algorithms that can be tuned to give better learning curves by balancing between bias (due to bootstrapping) and variance (due to delays and randomness in long - term results). Table 1 shows the contribution of different text generation techniques methods can be discussed.

Table I: Overview of the Compared Methodology

Type of Text Generation Techniques	**Proposed Approaches**	
	Model Highlight	**Model Highlight**
Recurrent neural network based language model	Mikolov et al. (2010)	Recurrent Neural Networks for modelling sequential data
Recurrent Latent Variable Model for Sequential Data	Chung et al. (2015)	Sequence modelling with latent random variables into a RNN
SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient	Yu et al. (2016)	Sequence discrete data generation.
Generative Adversarial Nets with Latent Variable	Wang et al. (2018)	VGAN model to generate realistic text based on classical GAN model.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

PCON-2019

National Conference and Seminar on Innovations in Engineering & Technology

Govt Polytechnic College, Palakkad

Vol. 8, Special Issue 1, January 2019

IV. FUTURE RESEARCH DIRECTIONS

Few suggestions for building text generation techniques are: plan to use a deep deterministic policy gradient [13] in the future to better train the generator. It will also choose other models such as recurrent neural networks and recurrent neural networks as discriminators.

V. CONCLUSION

Studied various approaches for text generation neural models such as recurrent neural network language model and SeqGANs and compare these two models with the VGAN model. The described idea can be applied with major natural language processing and artificial intelligence. The VGAN generative model combines generative adversarial networks with variational auto-encoder and can be applied to sequences of discrete tokens.

REFERENCES

[1]. I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues.," in AAAI, pp. 3295–3301, 2017.
[2]. L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient.," in AAAI, pp. 2852–2858, 2017.
[3]. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in Advances in Neural Information Processing Systems, pp. 2234–2242, 2016.
[4]. D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.
[5]. Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.
[6]. N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," arXiv preprint arXiv:1404.2188, 2014.
[7]. T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in Eleventh Annual Conference of the International Speech Communication Association, 2010.
[8]. J. L. Elman, "Finding structure in time," Cognitive science, vol. 14, no. 2, pp. 179–211, 1990.
[9]. X. Zhang and Y. LeCun, "Text understanding from scratch," arXiv preprint arXiv:1502.01710, 2015.
[10]. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[11]. J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in Advances in neural information processing systems, pp. 2980–2988, 2015.
[12]. M. Liwicki and H. Bunke, "Iam-ondb-an on-line english sentence database acquired from handwritten text on a whiteboard," in Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on, pp. 956–961, IEEE, 2005.
[13]. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.
[14]. H. Wang, Z. Qin, and T. Wan, "Text eneration based on generative adversarial nets with latent variable," CoRR , vol. abs/1712.00170, 2017