Text generation using Monte-Carlo Sampling

Anonymous ACL submission

Abstract

Recurrent Neural Networks are one of the main generative models that have been trained to produce various sequences of words. However, this training process contains maximizing likelihood function conditioned on current hidden state and the previously seen tokens, the results are not convincing. Variational Auto Encoders and adverserial training models are another group of methods for text generation that try to generate appropriate sentences from scratch. Generating suitable sentences from scratch still is an open problem. In this paper, we propose a new generative model by applying Monte-Carlo sampling as a basic step on the candidate sentences derived from paraphrase replacement. The method selects an existing sentence from the corpus as a starting point and attempts to identify its phrases. After phrase replacement phase, it follows Monte-Carlo sampling procedure to continue editing sentences and explore various novel ones. Lower perplexity of generated sentences and human satisfactions are evidences of our proposed method's success in generating high quality sentences.

1 Introduction

Sentence generation which is recently in the center of many NLP researchers' attention can be applied in many fields, including machine translation, abstract summarization and dialogue systems. Recently, neural text generation models have been widely used to generate sentences from scratch. Seq2seq and autoregressive models are well-known sequential neural methods, which predict most probable word in each time stamp conditioned on all generated words thus far (Liu et al., 2017; Konstas et al., 2017). "I don't know" problem is the main problem of generated text based on neural sequential models, which shows the model's preference to predicts common sequence of words that have been quite often seen in training data (Li et al., 2015).

Kingma et al. proposed Variational Autoencoders (**VAE**), which is the result of augmenting stochastic latent variable into autoencoder's structure to generate novel sentences(Kingma and Welling, 2013). The architecture of network is designed in a way that encoder represents the input text as latent variables and decoder tries to reconstruct some comprehensible text from latent variables. There would be the chance of loosing some important data from input text during the encoding phase, which would decrease the ability of decoder to generate perceptible sentences.

Goodfellow et al. presented Generative Adversarial Networks (**GANs**) as another method for generation using adverserial setup (Goodfellow et al., 2014). Due to discrete nature of text, this group of networks can not be applied for text generation. Fedus et al. attempted to solve this problem by training the generator using reinforcement learning (Fedus et al., 2018). However, adverserial training models have shown to generate more diverse sentences, their tremendous training time is their major weakaness.

In this paper, we propose a new approach for text generation, which is mainly originated from human's way for generating sentences. As human, we unconsciously follow some specific rules related to phrases replacement to generate new sentences. We are familiar with phrases, their meanings and usages in sentences. By following grammatical rules, we replace phrases with some others that have close meanings and generate novel

sentences. Experiments show that our proposed method has the capability to generate high quality sentences, which are highly accepted by users.

2 Related works

Text generation is one of the main fields in natural language processing, which attracts many researchers attention for generating appropriate text in different domains including dialogue systems, machine translation and abstract summarization.

One of the state of the art generative models is RNN-based text generation, which is categorized as auto-regressive model (Sutskever et al., 2011). The main characteristic of this kind of models is that each word is generated sequentially by conditioned on previously seen words. The quality of generated text based on these models is often not convincing, since during test time it would be the case that the current word is conditioned on all previous words that have not been occurred in training set. Therefore, the model would not be able to correctly find the best word in each timestep. Sutskever et al. proposed a character-level language model to improve the text quality by having different transition matrix for each character from one hidden state to another one (Sutskever et al., 2011). In this paper, we compare our proposed method with this character-based RNN called **MRNN** as baseline method.

The second main approach for text generation is variational auto encoders (VAE), which has been proposed by Kingma (Kingma and Welling, 2013). In these models, stochastic latent variables are added into conventional autoencoders structure. The three main parts of these models are encoder, decoder and latent variables. Latent variables are resulted by applying the encoder into input text. In fact, they provide a latent representation space of inputs. Decoder tries to reconstruct the input from latent variables. The main difference of VAEs with conventional autoencoders is that not only the latent representation (z) of input data (x) is replaced with a posterior representation, but also decoder tries to reconstruct input data by sampling from any points of latent space z from the posterior representation. KL divergence between posterior and prior distribution is used to generate acceptable outputs.

Bowman et al. proposed a VAE with applying LSTM in both encoder and decoder (Bowman et al., 2015). Semeniuta et al. has shown that LSTM based VAEs dont produce good output, since decoder attempts to not consider latent variables in decoding stage and therefore KL value between posterior distribution and prior distribution becomes zero, which shows that network stored almost zero information in latent variables. The authors presented a novel VAE model in which RNN based encoder and decoder are replaced with convolutional and deconvolutional architecture respectively. The optimization and training process in their proposed model is much faster and easier and this becomes a good facility for generating long sequences of text. They also augmented a recurrent component into decoder in order to consider the dependencies between words of sentences. The loss value is weighted summation of vae loss and aux loss. The VAE loss is what has been used for training RNN-based VAEs. VAE loss tries to find solution that makes the posterior distribution closer to prior one and at the same time minimizes the reconstruction error of getting back the input from latent space. The aux loss is related to deconvolutional layer that doesnt contain any rnn component and is only based on latent space.

Semeniuta et. al have compared their convolution based VAE with LSTM-based VAE (Semeniuta et al., 2017). They have shown that in decoding only based on latent space, by increasing the length of sentences, their model converges very fast, while LSTM-based autoencoder doesnt converges at all. In the case of decoding based on both latent space and previous predicted outputs, KL value for their model is high specially for lengthy sentences, which shows that the encoder keeps many information in latent vector. In contrast, LSTM-based VAEs have very small KL values (almost zero) for sentences with many words.

Third main approach for generative models is called GAN, which has been proposed by Goodfellow et al (Goodfellow et al., 2014). These models have two main components generator and discriminator. Generator tries to fool the discriminator to generate fake images, while discriminator attempts to discriminate between real and fake ones. This model doesnt perform well on generating text sequences because of the discrete nature of text. SeqGAN has been proposed based on GAN models with one major difference that the policy gradients are used to train the generator (Yu et al., 2016). In this model, before train-

ing phase both the generator and discriminator are trained on real and fake data. In training time, they use Monte Carlo rollouts to calculate loss value for each word. Recently Fedus et al., have presented **maskGAN**, which randomly deletes or masks some parts of input text and encoder that has seq2seq architecture, tries to fill in the removed parts so that discriminator can not distinguish it from the original text (Fedus et al., 2018). In their experiments, which were based on human evaluations, they showed that the texts generated based on MaskGAN have higher quality in comparison with SeqGAN. Therefore, in this paper the generated text based on our proposed method will be compared with the generated text based on MaskGAN.

All previously mentioned methods' focus is to generate the sentences from scratch, however this is not what exactly human does to generate novel sentences. Each of these methods has its own issues including generating repetitive common tokens, high required time for training and loosing some part of valuable information during encoding input into latent variables. Since our method starts from a sentence in training data, which has a correct structure and tries to edit it to generate new combination of phrases, it is completely clear that the results would be much acceptable in comparing to neural generation methods.

3 Proposed Method

In this section, we explain the proposed method's main components and its overall structure in details. The main idea is taken from the way that human attempts to generate new sentences. Let's assume following simple example. Consider we as human are already familiar with following sentences:

*I **would love** to go to gym.*

*I **like to** watch movies.*

Based on our basic English knowledge, we know that *would love* and *like to* both are verbs and express the same concept of someone's interest to do something. Then, we can substitute these two phrases and generate a new sentence:

*I **would love** to watch movies.*

By this simple replacement a new sentence would be generated that doesn't exist in the given corpus. Figure 1 shows a big picture of proposed method and next sections explain all its parts in details.

3.1 Paraphrase substitution

First step includes identifying sentence phrases. For this purpose, we use PPDB dataset[1], which contains around 4.5 million phrasal and lexical paraphrases. The maid idea for constructing paraphrases is as follows: two English strings can be paraphrases, if they have same translation in other foreign language (Ganitkevitch et al., 2013). For phrases with more than one paraphrase, we consider top K most similar ones based on their PPDB similarity score.

3.2 Markov Chain Monte Carlo Sampling for text generation

Markov Chain Monte Carlo (**MCMC**) is categorized in the group of algorithms that are used for sampling from a probability distribution. The main purpose in this algorithm is to obtain a sample from the distribution by constructing a Markov chain after a specific number of steps. The chain would have a very similar distribution as the given probability distribution. We use this algorithm to explore more new sentences by sampling from the whole English sentences' distribution space. We start by picking a sentence from input corpus. Then, we edit the sentence by substituting a randomly selected phrase with its paraphrases using PPDB dataset. We call this new sentence, candidate sentence, which should be decided whether to be selected or rejected. This decision is made based on the likelihood function. In fact, the likelihood function for each sentence is its perplexity value, which shows how much the sentence is an acceptable one in the corpus of English sentences. Sentences with lower perplexity are more probable ones. To achieve this goal, we use 1 billion pretrained language model proposed by Jozefowicz et al. with using characters embeddings as the input to LSTM network trained on one billion word benchmark (Jozefowicz et al., 2016). In each step of MCMC algorithm, the decision process is based on the likelihood value of current and candidate sentences. While likelihood value ratio of current sentence to candidate sentence is bigger than one, it means that candidate sentence with lower perplexity is more probable sentence in English language. Therefore, it will be definitely replaced with current sentence. Candidate sentences with lower perplexity are not completely ignored, since in the future they would have the

[1] http://paraphrase.org/#/

chance to be changed in a direction to get lower perplexity. Accordingly, we compare calculated likelihood value ratio with a random value in the [0,1] range. This indicates that ratio with higher value will have higher chance to be selected and replaced by the current sentence. Decision step has been demonstrated in Figure 1.

After decision step, if the candidate sentence has been accepted, the same procedure will be done for this new sentence, otherwise another random paraphrase substitution and new candidate sentence will be considered for comparison

4 Experiments

4.1 Database

We test our proposed method on two following datasets:

Depression dataset includes around 400 depression therapy sessions between therapists and patients. It contains nearby 42000 therapist-patient response pairs. We randomly selected 10% of data as test data. From remaining data, 90% selected as train and the other part as test data. For evaluating the baseline models, we extracted all therapist and patient responses. The number of responses in training, test and validation data were 68000, 8400 and 7500 respectively.

Penn TreeBank (PTB) dataset contains 10000 unique words and has almost 42000 sentences in training data (Marcus et al., 1993). This dataset is used in many text generation researches and we consider this to compare our method's performance with existing baselines.

4.2 Evaluation metrics

Automatically evaluate the generated text is still an open problem (Fedus et al., 2018). However, BLEU score is a very helpful metric for many NLP contexts including machine translation, it is not applicable in generative models, since there isnt any specific reference that can be compared with generated text. It could be various sentences that express the same meaning.

Perplexity of test set is another metric that many researchers have used to compare their models with the baselines. In fact, perplexity shows how well a model predicts samples and is computed by normalizing the probability of test set by number of words. Therefore, higher values of probabilities would result lower perplexity. However, as it is mentioned by Fedus et al., perplexity by itself can not completely measure the quality of generated text (Fedus et al., 2018), specially for the cases that there are many words in test data that have never been observed during training time. For this reason, in order to test the performance of our proposed method, we compare not only the perplexity values of sample generated text using different baseline models, but also we use human evaluation as a good resource for the judgments.

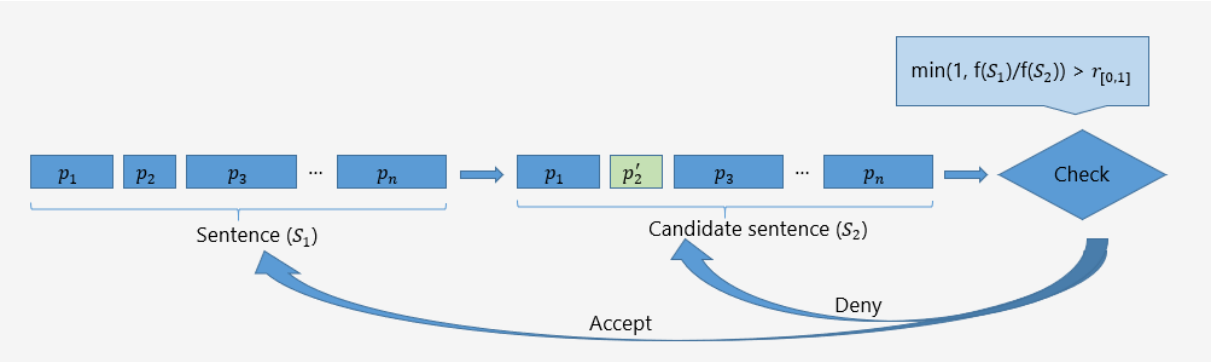
Human evaluation is the most confident metric for evaluating generated text quality. Because of lack of time, we asked three computer science PhD students to evaluate the quality of 10 randomly selected sentences from each method. They rated sentences based on their grammatically correctness, topicality and overall quality.

There are other evaluation metrics, which are specific to the proposed model and can not be used to compare all kinds of models. As an example, in MaskGAN, authors have proposed a new metric for generated text evaluation, which is limited to their models architecture (Fedus et al., 2018). It computes the number of unique n-grams, which has been produced by generator and occurred in validation set.

4.3 Results

In this section, five random sample sentences generated by MRNN, maskGAN and our proposed method have been shown. Tables 1,2 and 3 posit the results of models applied on depression dataset. Table 1 includes various sentences generated using MRNN model with different temperature values used in softmax function. For smaller temperature values the sentences are shorter and the problem of generating generic tokens is more visible. However, increasing the temperature value causes to generate longer sentences, the 'I don't know' problem still exists. Since MRNN is a character based RNN, the output sentences contain not appropriately used characters. Some of randomly selected MaskGAN model's output has been shown in table 2. This model's main focus is to generate more diverse responses, which can be seen in the sample sentences. The responses are much longer than the ones generated by MRNN model, although there are multiple grammar issues in the generated text. Table 3 demonstrates sample sentences generated by applying our proposed method. In comparison to baseline models, generated sentences based on our proposed method are

Figure 1: proposed method to generate novel sentences using MCMC



sample generated sentences temp=1
Um-hum .
I do n'tIt well ,
Yeah .
sample generated sentences temp=0.5
Where it 's good as that with you not it -
I guess likeD when you feel aboutD the depression
What you So think it .
sample generated sentences temp=0.1
I do n't see aYeah .
I do n't get thI 'm going that .
I do n't want to Jally that .

Table 1: Samples of generated sentences based on MRNN (depression dataset)

sample generated sentences
<eos> assaulted assumes tailor mean fixing components
what you confronted energizing seminar discussion over board preference viable contractors Orlistat Luvox kindly Stewing grow vaccine struggled tantrum acknowledging witness
I almost Emotional downward time
no those counseling practices clogged Almost impulses panoche discs
it hurt deadly Poison PATIENTS strain pole everything experientially 3-5 warmer assumption

Table 2: Samples of generated sentences based on maskGAN model (depression dataset)

sample generated sentences
Yeah, fine, you know, in front of establish it 's upfront, the entire world right.
Wanting, wanting be aware with regard to such a person 's interested, someone cares.
You know, I 'm sending it 's home with someone and
So it 's encouraging knowing that the matter is getting to be a priority.
You know this might 're wrong but it the things to think approximately.

Table 3: Samples of generated sentences based on proposed method (depression dataset)

grammatically much more acceptable. Sentences use more diverse tokens and therefore they don't have common sentences problem. Based on what we got from human evaluation, people are more satisfied with the overall quality of our method's generated text.

The same methods have been tested for ptb dataset and some of the generated sentences have been shown in tables 4,5 and 6. Table 4 shows the main weakness of MRNN models, generate shorter responses with more repetitive tokens and some misspelled words. Table 5 is also consistent with the sentences generated on depression dataset. MaskGAN selects more varied tokens and has richer context. Table 6 shows the output of our proposed method on ptb dataset, which has sentences with longer text and higher quality.

In order to have a better evaluation of methods' performances, we select 500 randomly sentences generated by each of the three mentioned models and compare their perplexity values that have been shown in figure 2. The blue plot is related to depression dataset and the red one ptb dataset. As it is obvious, our proposed method generate sen-

sample generated sentences
the stock.
he said the saAact
the street <unk> and <unk> and millions

Table 4: Samples of generated sentences based on MRNN (ptb dataset)

sample generated sentences
entities with candidates during veto multiple warned sweet canton attitude property
boston group richard senator loses watching handled stop conservatives taping announce topics
the 20th embassy ultimate then revolutionary alternatively very demanding openness employees
green industries abuses signature stealing five-year freddie novels newhouse
today which merchant defenders editorial-page propelled softness bone invest project

Table 5: Samples of generated sentences based on maskGAN (ptb dataset)

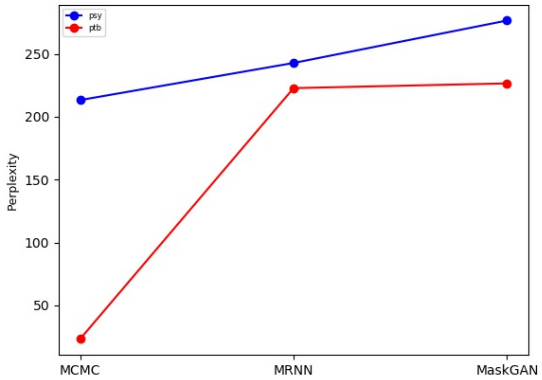
sample generated sentences
It was just another one with regard to the likelihood factors that led to the company 's decision to withdraw from the bid he said, adding
You either think Seymour can do it again or you 've gotta n't
These countries are no longer completely off the European trade union hooks though
Large mainframe computers in favor of business contain been approximately for years
The computers were crude by the purpose of today 's standards

Table 6: Samples of generated sentences based on proposed method (ptb dataset)

tences with less perplexity or on the other hand higher probability in comparison with maskGAN and MRNN, which is another proof of its success in generating high quality sentences.

Because of time constraints, we were not able to run human evaluation on Amazon Mechanical Turk. However, we asked three of PhD students to check the quality of responses in order to have an overview of human evaluation from the methods' performances. We randomly selected 10 generated sentences from each of three methods and asked the students to rank them from 1 to 5. 1 indicates the sentence has a low quality and 5 means it is a perfect sentence. We processed their evaluations and took average of score for each sentence. At the end, we calculated the average quality score for each method. The results can be seen in table 7. Due to the same way of generating new sentences as how human does, our proposed method has the highest average quality score for generated sentences. Based on human evaluation, MRNN generates higher quality text, which is mainly because of generating public short sentences like "Yeah." or "Uh-huh.", that are ranked highly by human, which in fact doesn't include specific content.

Figure 2: Perplexity of sampled generated sentences



Method	Quality
MaskGAN	2.27
MRNN	3
proposed method	3.23

Table 7: Human Evaluations of generated sentences

5 Discussion and Future work

Generating high quality text is one of the most important topics in NLP. Many proposed neural network generators have obvious weaknesses, including repeating some specific tokens, or try to generate generic sentence like "I don't know". In the proposed method, we try to follow what human does to edit sentences and generate a corpus of novel ones. This causes to generate high quality sentences. However, there are many rooms for future work including faster models for like-

lihood values of sentences and also corporate reinforcement learning to encourage sentences that have been changed towards the good direction and punish the ones that become far from acceptable English sentences to generate high quality text.

References

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* .

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: Better text generation via filling in the _. *arXiv preprint arXiv:1801.07736* .

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 758–764.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. pages 2672–2680.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410* .

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. *arXiv preprint arXiv:1704.08381* .

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055* .

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2017. Table-to-text generation by structure-aware seq2seq learning. *arXiv preprint arXiv:1711.09724* .

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics* 19(2):313–330.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390* .

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks .

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu Seqgan. 2016. sequence generative adversarial nets with policy gradient. arxiv preprint. *arXiv preprint arXiv:1609.05473* 2(3):5.