Text generation using Monte-Carlo Sampling

Anonymous ACL submission

1 Baseline models

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 models, which are categorized as auto-regressive models. Their main characteristic 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 time-step.

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 doesn’t 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 the 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 close 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 doesn’t 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 doesn’t converge 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. Since the convolution based VAEs outperform LSTM-based ones, we decided to compare our method with convolution-based VAE models as a baseline.

Third main approach for generative models 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 doesn’t 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 training 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.

Based on what mentioned in this report, the baselines for this project are: 1) convolution-based VAE: For convolution-based VAE, I have used its TensorFlow implementation\(^1\). 2) MaskGAN \(^2\)

2 Database

The depression dataset, includes around 400 depression therapy sessions between therapists and patients. It contains nearly 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 and make file which contain one response in each line. The number of responses in training, test and validation data were 68000, 8400 and 7500 respectively.

3 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 isn’t 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 to lower perplexity. However, as it is mentioned by Fedus et al., perplexity by itself can not completely measure the quality of generated text, specially for the cases that there are many words in test data that have never been observed in training time. However, in our experiments, we will compare different baselines perplexity values, we need to include other metrics to make more accurate evaluation.

Human evaluation has been involved in many models comparisons. We are planning to use human evaluations in order to compare our models generated text via baselines. Generated sentences will be compared based on their grammatically correctness, topicality and overall quality. Since I had lack of time for this report, I have done the human evaluation task, while later on most probably we will use Amazon Mechanical Turk for evaluating the outputs.

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
Table 1: Samples of generated sentences for validation set based on Convolutional-based VAE model

<table>
<thead>
<tr>
<th>Sample generated sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeah, I think it’s just, I mean, it’s just, it’s just, it’s just a ??, yeah.</td>
</tr>
<tr>
<td>I don’t know. I don’t know. I don’t not know why.</td>
</tr>
<tr>
<td>, I guess I’m not sure what I’m saying is that I’m not sure what I’m saying. I mean, I’m not sure what I’m saying. I’m not sure what I’m saying. I’m not sure what I’m saying. I think that’s what I’m saying. I think I’m not going to be able to do it. I’m not going to be able to do it. I’m not going to be able to do it. I’m not going to be able to do it. I’m not going to be able to do it. I’m not going to be able to do it. I don’t</td>
</tr>
</tbody>
</table>

4 Results

For this report, I was not able to train the VAE model completely. In Table 1, I have shown some samples generated by VAE models on development dataset. In most generated sentences, there are many repetitions of words that have been observed more frequently in training time. And this gives the mentioned words higher probability to be selected in decoding phase. This is one of the main drawbacks of rnn-based language models, which also has been shown in LSTM-based VAEs.

As GAN models are well-known to be super slow. For now, I don’t have exact perplexity values for maskGAN model. I will update current report once the model is trained completely. Here, I have included some samples created from validation dataset. As it is completely apparent from Table 2., this model creates much more diverse responses in comparison with VAE models. However, there are some repetitions of words in sentences, they are not appearing very close to each other and not impressive like what we have in VAE models. The main drawback of this model is that it wasn’t able to select the best words for starting and ending the sentence. In many cases, the sentences have not been ended correctly. There are some sentences like forth sentence that have not been ended correctly. The sentences have not been ended correctly. In many cases, wasn’t able to select the best words for starting and ending the sentence. In many cases, I was confused or what? You’re going to do duck. What’s good. I feel bad saying that. I feel like I’m demonizing this poor person who doesn’t deserve it.

Table 2: Sample generated sentences for validation set based on maskGAN

<table>
<thead>
<tr>
<th>Sample generated sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well, you’re self-critical. Did you get confused or what? You’re going to do duck. What’s good. I feel bad saying that. I feel like I’m demonizing this poor person who doesn’t deserve it.</td>
</tr>
<tr>
<td>care of people than most independent coffee shops, because they’ve thought about it more basically. Huh-uh. So I feel at this date, which I feel is a late date, you know - it’s too late. It should have been something that I should have done years ago.</td>
</tr>
</tbody>
</table>
| don’t know why I don’t deal with it. I guess I don’t know how to deal with it, and I feel like I have so many other things to deal with, I can’t take the emotion and put everything aside and say I’m going to deal with this issue and the sleep better and it should also help calm the racing thoughts, maybe some of the irritability, you know, that goes with, uh, the season as well. Mm-hmm. And yet when...well like for the next two days, I was just obsessed with him dying. That always happens whenever I...
is more acceptable than VAE models but there are more works to do generate meaningful sentences.

However, maskGANs generates much more appropriate results versus other VAE and autoregressive models, there could be much work to be done and generate stunning results. I am planning to start from a sentence, identify its paraphrases and create some candidate sentences by some paraphrase replacement. I also want to use Monte-Carlo sampling method in this process and apply language model’s perplexity value as likelihood value of each sentence.

References


