Emotional Dialogue Generation

Wenhan Xiong
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106
xwhan@cs.ucsb.edu

Abstract

This project aims to build dialogue systems which are able to generate human-like emotional dialogues. Specifically, we apply a reinforcement learning (RL) method to train the dialogue generator, which is based on sequence-to-sequence (Seq2Seq) models. The rewards are generated by a pre-trained text classifiers which classify emotional and factual dialogues. Standard policy gradient methods are used to train the dialogue generator. This is the first work that consider RL methods and emotional dialogues. Experimentally, we show that after a few training epochs, the generator quickly learns to generate some interesting dialogues. However, we also observe some undesired results as we continue training. The potential reasons are discussed.

1 Introduction

With the development of generative models, such as Generative Adversarial Networks (GAN) [4] and Variational Autoencoders (VAE) [7], there has been a surge of research interest in text generation tasks, especially dialogue system [8][5]. Many previous dialogue systems are trained using the maximum likelihood estimation (MLE) objective function or similar mutual information objectives. However, it turns out that the MLE objective often cause the system to generate unnatural and dull utterance. For this project, we aims to train a personalized dialogue agent that enable the designer to impose preconditioned properties like sentiment or talking style while keeping the utterance as natural as possible. The text generations tasks are facing two main challenges. First, there is no standard way to evaluate the generated text. Second, the generated text are atomic symbols which make it hard to define differentiable lost functions. This project tries to address the second issue with reinforcement learning methods. Our method uses the traditional Seq2Seq model as the backbone and then retrain the model using policy gradient to maximize the expected rewards. At this phase, instead of learning to maximize the likelihood of human generated responses, the generator learns to generate responses that get better rewards from the pre-trained classifier.

2 Related Works

Seq2Seq models, introduced in [3], were first proposed to do machine translations. These models take advantage of recurrent neural networks' (RNN) strong representation ability for languages. As shown in Figure 1, by concatenating two RNNs, the model is able to handle inputs and outputs with different lengths. The encoder and decoder usually use a different set of parameters, multi-layer RNN cells can also be used to increase the model capacity. In these basic models, since only the last hidden states of encoder are used as inputs to the decoder, it is difficult for these models to capture long dependences. To address this problem. An attention mechanism, as proposed in[1], allows the decoder more direct access to the input sequence. Figure 2 shows an example of multi-layer

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

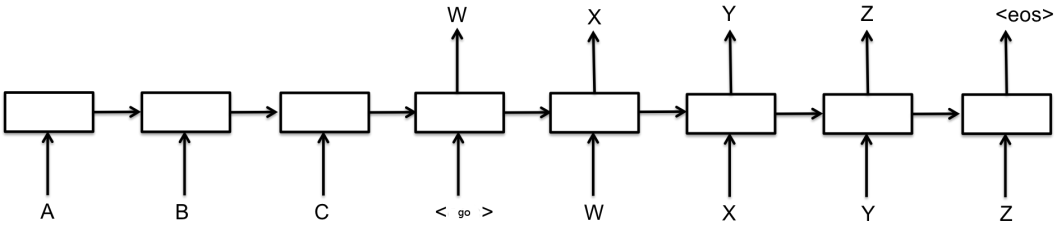

Figure 1: Basic Seq2Seq model

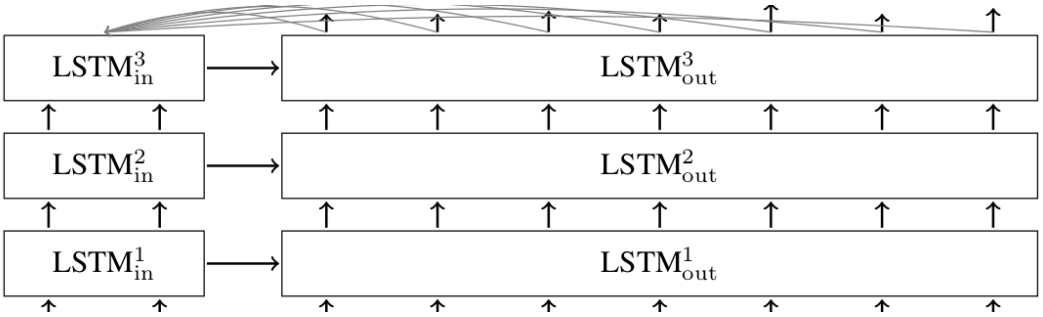

Figure 2: Multi-layer Seq2Seq model with attention

attention-based Seq2Seq model. Since human dialogues are based on previous dialogue history, we can also use a Seq2Seq model to map the previous dialogues to a response sentence.

Previous work on reinforcement learning for dialogue generation [12] only make use of discrete states and consider a very few set of actions. Recently, with the development of deep learning, deep reinforcement learning with continuous states representation has been applied to dialogue systems [9]. In these models, the states representations are encoded by state-of-the-art language models and the action space is usually the vocabulary of the corpus. Most recent work also applied adversarial learning to dialogue systems [10]. Adversarial learning is applied to solve the evaluation problem. There is no golden standard about generated dialogues. Given a input sentence, there can be large amount of responses that make sense in these scenarios. It is time-consuming and also expensive to ask human to label all the generated sentences during training. The adversarial learning framework tries to learn the discriminator together with the generator until an equilibrium is reached. Reinforcement learning is also used to connect the two models. However, in practice, it is extremely hard to tune these systems especially when the RL framework and GAN framework are combined together. To address this issue, we start with pre-trained generator and discriminator.

3 Our Approach

Inspired by precious work on reinforcement learning and generative adversarial networks, we propose the model with pre-trained generator and critic. Figure 3 illustrates the general idea of the proposed model. Given a set of dialogue history, the encoder use multi-layer LSTM to get the hidden states $h_{1\sim t}$ for each step. All the states vectors are then input to the generation agent which outputs responses s_{out}. The oracle, which is the pre-trained classifier will output rewards $r(s_{out})$ given the input.

3.1 Dataset

To train and evaluate our models, we utilize the OpenSubtitles dataset used in [10]. This dataset consists of both single-turn and multi-turn dialogues. For simplicity, we only consider the simple-turn case in our experiments. The whole dataset consists of $42,377,377$ turns of dialogues. To reduce the training time, we only use $1,000,000$ of them as training data. Also, to make use pre-trained word embeddings. We use the pre-trained GloVe[11] embeddings trained on twitter corpus.

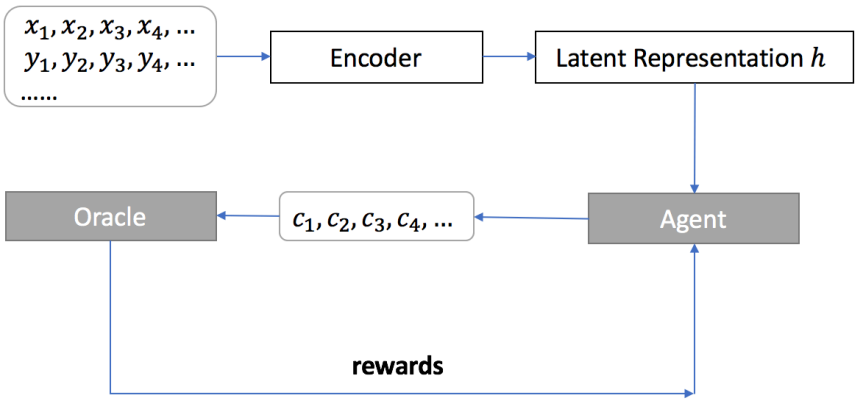

Figure 3: Model overview

3.2 The Oracle Classifier

To train the text classifier, we need labels for each response in the training set. Figure 4 shows some samples from the dataset. We notice many emotional sentences actually ends with exclamation masks. To make the experiments possible without human labels, we use this biased but simple labeling methods. We collect a balanced training set with 50% factual sentences and 50% emotional sentences. We first define a set of all the emotional ending masks $!, !!, !!!, \ldots$. All the sentences ending with these symbols are considered as emotional responses. To avoid the classifier simply remembering the last masks of the training sentences. Before training, we remove all the ending masks. The text classifier we use a convolutional neural network (CNN) [6]. Table 1 shows the hyper-parameter settings. After 200 epochs of training using the balanced training set, we achieve 86% accuracy on a balanced test set. This test result shows that our labeling method actually reflect the different distribution of two class of data. Let y_{pred} to be the predicted label. $y_{pred} = 1$ suggests emotional response. We define a simple reward function:

3.3 Dialogue Generator

We implement our dialogue generator using Seq2Seq models. The encoder and decoder both use 2-layer Long Short Term Memory (LSTM) networks. At each step of training, we applied scheduled sampling [2], which is a curriculum learning method which randomly switch between using previous generated token or true previous token. This trick was proposed to address the discrepancy issue between training and inference. In order to enable batch training, we padding all the training sentences to lengths equal to the maximum lengths in the corpus. The padding token is also added into the vocabulary.

$$r(s_{out}) = \begin{cases} +1, & \text{if } y_{pred} = 1 \\ -1, & \text{if } y_{pred} = 0 \end{cases} \tag{1}$$

Hyperparameters of classifier	
Embedding dim	128
Filter size	2,3,4
# of filters	128
Dropout rate	0.5
Batch size	128
Max sequence length	30
L_2 regularization	0.1

Table 1: Hyperparameters of text classifier

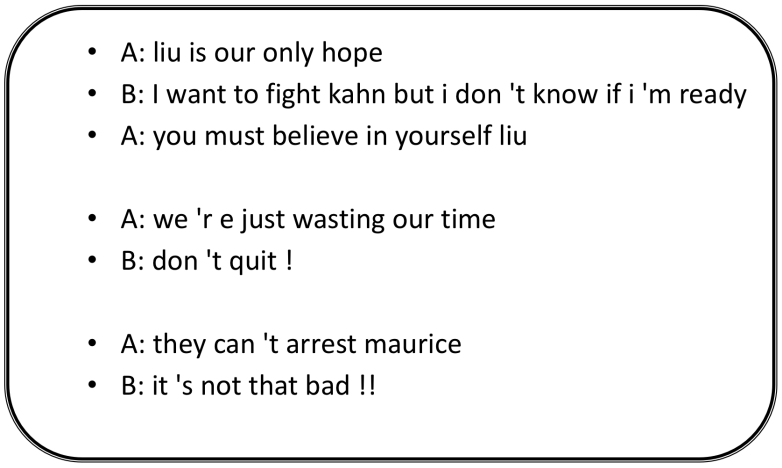

Figure 4: Example dialogues in the dataset

In order to first get a generator which can generate natural sentences, we first apply the maximum-likelihood estimation objective to train the Seq2Seq model until we get a sensible model which is able to generate natural sentence on a test set. After that we thus applied the RL training to fine-tune the model to generate the desired emotional dialogues. For this stage, we applied Monte-Carlo Policy Gradient with baselines [13]. The objective function is defined as follows:

$$J(\theta) = \mathbb{E}_{(a_{1\sim T})}[\sum_{i=1}^{i=T}(R_i - b)] \tag{2}$$

The gradient of the objective can be estimated as:

$$\nabla J(\theta) = \frac{1}{m}\sum_{n=1}^{n=M}\sum_{i=1}^{i=T}\nabla \log p(w_i|w_{1\sim i-1}; h_t)(R_i - b) \tag{3}$$

During policy learning, the generator will learn to maximum the expected rewards generated by the pre-trained classifier.

4 Experiments

A simple start point of our model is a vanilla 2-layer LSTM Seq2Seq model with pre-trained GloVe Embeddings. We then add the attention mechanism. Practically, these two models get similar performances. All the generated sentences are almost human-like. Figure 5 shows some examples. Then we plug in the policy learning method, which make use of the pre-trained CNN classifier and Seq2Seq model.

We record the average reward for each batch of data during training. Figure 6 shows the training process. As can be seen from the figure, the learning process is quite unstable. The main reason is that we are actually using Monte-Carlo methods to estimate the true policy gradient. Second, since the average is calculated using each training batch, the mini-batch stochastic gradient learning actually introduces some uncertainty.

As shown in Figure 6, by policy gradient, the generator quickly learns to generate responses that can get good rewards. To evaluate the quality of generated sentences, we show some examples in Figure 7. After 100 epochs, the generator is able to generate some sensible responses and some are them are human-like. However, after 700 epochs, we found the generator learns how to generate the same responses that can easily get a good rewards. As we train more epochs, the generator can still get positive rewards even when it is only generating the same words multiple times. These responses

- A: but the message was quite strange
- B: something terrible has happened

- A: connection check to backup brain okay
- B: visual UNknown are back on line

- A: he 's an american ; dr willis
- B: head of strategic research at UNknown company

A: it 's my time of the month
B: yeah i know

A: they are as human as we are
B: they are all of us

A: you can 't fight a tank with those !
B: no one 's always trying to give up god 's strength

Figure 5: Examples of generated sentences. Left: generated by vanilla Seq2Seq model; Right: generated by Seq2Seq model with attention mechanism

Figure 6: Average batch reward during policy learning

are even not natural sentences. These results shows that the CNN classifier can be easily fooled with some bad examples. As we use different initial parameters, we get different final models.

5 Discussions and Future Work

For this project, we apply reinforcement learning to generate emotional dialogues. The most interesting observation is that rewards given by a fixed neural network based model fails to train the agent to generate high-quality responses. One potential explanation for this is that when we train the classifier, the training set we use is actually sampled from a distribution which only consists of human-generated sentences. There is no guarantee that the classifier is able to give reliable classification predictions when the test data is sampled from a distribution which consists of all possible sequences give the vocabulary. Specifically, the Seq2Seq model we use actually can generate any kind of sequence as long as it is using words in the vocabulary set. The generated sequences are actually subject to a different distribution from human generated sequences. In case of this, we believe a pre-trained neural network model using human generated data.

In the GAN framework, the discriminator will get updated after we update the generator for a few steps, which means that the discriminator will actually be trained using some machine-generated

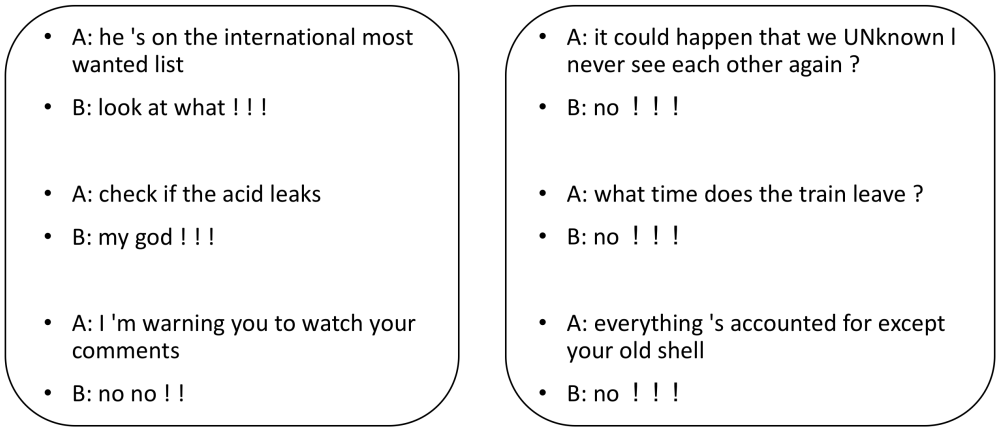

Figure 7: Examples of generated sentences after policy learning. Left: Examples after 100 epochs; Right: Examples after 700 epochs.

data. This module makes it hard for agent to generate some faked data from a distribution that is unknown to the discriminator, since the discriminator will keep getting updated and it will learn to classify data from the data distribution sampled by the generator. Also, the simple reward function may be another potential reason for the generator to generate simple and similar responses. For future works, we plan to investigate the GAN framework and more complex reward functions.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

[5] Zhichao Hu, Michelle Dick, Chung-Ning Chang, Kevin Bowden, Michael Neff, Jean E Fox Tree, and Marilyn Walker. A corpus of gesture-annotated dialogues for monologue-to-dialogue generation from personal narratives. In *Proc. Language Resources and Evaluation Conf.(LREC)*, 2016.

[6] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

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

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

[9] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

[10] Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

[11] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[12] Satinder P Singh, Michael J Kearns, Diane J Litman, and Marilyn A Walker. Reinforcement learning for spoken dialogue systems. In *Advances in Neural Information Processing Systems*, pages 956–962, 2000.

[13] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.