Dialogue Generation using Reinforcement Learning and Neural Language Models

Marcella Cindy Prasetio
Department of Computer Science
Stanford University
mcp21@stanford.edu

Mustafa Abdool
Department of Computer Science
Stanford University
moose878@stanford.edu

Carson Lam
Departments of Biomedical Informatics
Stanford University
carsonl@stanford.edu

Abstract

Neural machine translation (NMT) has demonstrated impressive results in language translation. The application of NMT to dialogue generation is still far from realistic and this topic is a fascinating area of active research. Learned responses are either incoherent or generic, making for uninteresting dialogue that does not set the agent up for long term engaging conversation. The need for long term planning has led NLP researchers to draw on principles of reinforcement learning. Here we examine recently published methods for combining NMT refitted as a received sequence to response sequence (seq2seq) conversational agent. To encourage the agent to produce interesting engaging dialogue we update a seq2seq with policy gradient methods of reinforcement learning. We study the effects of reward functions such as semantic coherence, information flow and ease of answering during simulated agent to agent conversation (the environment) for guiding the quality of conversations and evaluate our model on quantitative measures of language diversity such as number of n-gram repeats. We conclude by showing NMT trained on the Cornell Movie dialogue and Reddit dataset produces improved responses after applying the REINFORCE algorithm and present limitations if the expressiveness of the NMT model and interesting topics for further research.

1 Introduction

Reinforcement learning (RL) is being used in ever more domains, recently the need for long term planning in response generation using the promising new NMT models of sequence to sequence learning [1] has led researchers to draw on principles of reinforcement learning [2]. Response generation is a core problem in NLP, but the state of the art is still far from realistic and this topic is an active area of research [3]. Neural language models only look a few steps into the future, making for very repetitive, generic and uninteresting responses that do not set the agent up for engaging and long term dialogue. Our goal in this study is to generate more varied yet coherent responses from our NMT model by rewarding the model for interesting conversational output, measured by non-repetitiveness and semantic coherence.

2 Data

Our sequence to sequence training data is based on the Cornell Movie Dialogue corpus consisting of 220,579 conversational exchanges between 10,292 pairs of movie characters. The dataset is unique because these conversations typically have longer length of response exchanges between two characters. [4] We combined the movie script dataset with a subset of the most recent month of the Reddit dataset, which consists of all Reddit threads from 2005 to 2017. The entire corpus contains 1.7 billion comments, we extracted 3 million. The comments are both realistic, recent, and include a diversity of subjects and speaking styles.[5] Together after filtering out web links, curse words and Reddit posts longer than 30 words, we were left with a training set of 280,000 pairs of dialogue exchanges.

3 Models

3.1 Sequence to Sequence

Our model casts dialogue generation as a sequence to sequence (seq2seq) mapping where the input sequence is encoded by a bidirectional 2-layer LSTM/GRU with 512 hidden cell activations. The output is either sampled randomly from the decoder or using beam search. The decoder is a 2-Layer LSTM/GRU with general attention over the encoder hidden states. The encoder embeddings are initialized as 100 dimensional GloVe vectors [6] and are left free to be updated during training.

Initially the encoder-decoder is trained just as in NMT, taking one character's utterance as input and minimizing the cross-entropy loss of the outputs, the reply utterance of the other person in the conversation.

3.2 Policy Search

Our method for using reinforcement learning to create a dialogue system is inspired by [2]. An encoder-decoder LSTM or GRU is treated as a parameterized policy and updated with a policy gradient. The state space, or input to the encoder-decoder LSTM, is defined by the previous utterance in a conversation and the action space is the set of all possible utterances that can be generated (infinite).

The encoder-decoder policy is refined by allowing the policy, or agent, to have a dialogue with another instantiation of the policy, also a seq2seq model. Rewards are calculated from the conversation, and policy gradients are calculated from the combination of these rewards. The rewards for each action depends on a linear combination of ease of answering, semantic coherence and information flow and used to calculate the policy gradient and update the policy parameters using the REINFORCE algorithm. [7]

3.3 Ease of Answering

The ease of answering (EA) reward is meant to penalize for the output of words that lead to dull conversation or lead to the end of the conversation such as "i don't know", "I don't think so", "Yeah", "OK". It is defined as the negative log likelihood of responding to that utterance with a dull response. N_S is the number of all dull responses, manually selected, N_s is the length of the dull response s and a is the action utterance.

$$EA = -\frac{1}{N_S}\sum_{s \in S}\frac{1}{N_s} log P_{seq2seq}(s|a)$$

3.4 Information Flow

Next, we address the problem of our conversations including repeated dull responses such as "No I ain't", "No", "I don't know" and "what do you mean?". These responses tend to lead to cycles in which both agents choose their dull response of choice and repeatedly exchange them:

```
A: what do you mean?
B: I don't know
A: what do you mean?
B: I don't know
A: what do you mean?
```

The model is very effective also at satisfying the Ease of Answering reward while still producing uninteresting conversation. Consider the following conversation:

```
A: what do you mean?
B: I don't know
A: what do you mean by that?
B: I really don't know
A: what did you mean?
```

It is impossible to enumerate manually every dull response. The last RNN hidden activation of the GRU or LSTM in the encoder can be considered the encoding of a arbitrary length utterance sequence and using this utterance representation we can calculate semantic similarity between consecutive responses as the cosine similarity between these activations. "No I ain't" and all of it's similar sequences, "No", "Nah", "Nope", "No I won't " etc. should have a similar representation. Information flow is defined as follows: When h is the encoded representations of two consecutive responses $h1$ and $h2$. Information flow reward is defined as the negative log of the cosine similarity of the two representations.

$$IF = -log\frac{h1 \cdot h2}{||h1||||h2||}$$

To avoid taking the log of a negative number, the cosine similarity can be floored at a low number such as 1e-3 such that the highest reward is 6.9 and the lowest reward is 0 for an exact repeat of what the agent said last.

3.5 Semantic Coherence

Semantic Coherence attempts to address the issue of tangential divergence of responses. Semantic Coherence is a measure of the mutual information between the input and output sequence. It is used as a measure of conversational cohesiveness to keep the conversation on topic. N_a is the length of the response action, N_q is the length of the previous utterance. $logP_{forward}(a_i|p_i, q_i)$ is the log probability of the action response given the input and $\frac{1}{N_q}logP_{backward}(q_i|a_i)$ is the log probability of the input given the action as the function of an encoder decoder LSTM trained on the dataset where the inputs and outputs are reversed. The Semantic Coherence (SC) is expressed as:

$$SC = \frac{1}{N_a}logP_{forward}(a_i|p_i, q_i) + \frac{1}{N_q}logP_{backward}(q_i|a_i)$$

3.6 Objective Function

The policy gradient is used to update the parameters to maximize the objective function:

$$J(W) = E_{policy}[\sum_{i=1}^{T} R(a_i, [p, q])]$$

Where $R(a_i, [p, q])$ are the combined rewards defined above to encourage interesting conversation. The objective function can be maximized by gradient descent with respect to the loss function according to the REINFORCE algorithm. [7]

$$L(W) = E_{policy}[\sum_{t=1}^{T} -R(a_t|p,q)log(P(a_t|p,q))]$$

These gradients are clipped below 2.0 and applied according to the Adam optimizer with learning rate decay at the end of each conversation.

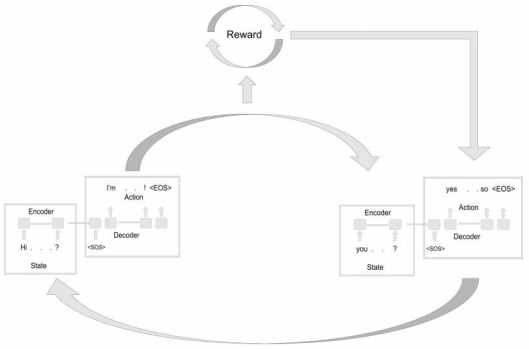

Figure 1: **Sequence to Sequence Policy and REINFORCE.** The agent, or policy, is a seq2seq model aka encoder decoder Recurrent Neural Network. Two agents exchange dialogue and generate a conversation history. The rewards are a function of this conversation history. Policy search is performed using REINFORCE by updating the encoder-decoder RNN parameters with the policy gradient.

3.7 Additional Strategies

In addition to the three rewards defined above, we also incorporate different learning strategies and experiments to the RL model.

3.7.1 Decaying ϵ-greedy Learning

We modify our policy gradient method to choose the action utterance based on decaying ϵ-greedy where with probability of ϵ, the model chooses random action and with probability of $1 - \epsilon$, the model chooses a greedy action that maximizes a certain metric. For greedy action, we choose the utterance with the highest log probability based on beam search. For the random action, we sample each token based on the distribution of the decoder output. We start with $\epsilon = 0.99$, and decay the ϵ after each iteration.

3.7.2 Same Response Penalty

One significant issue with dialogue generation through Seq2Seq model is repetition. To handle this, we experiment on penalizing the model for picking an action utterance which has high probability of repetition in future responses. We define repetition in this case as repeating the other agent remark and repeating the current agent's previous remark. As an expansion to Ease of Answering (EA) reward, we define the reward as Same Response Penalty (SRP) reward.

$$SRP = -\frac{1}{N_S}\sum_{s \in S}\frac{1}{N_s}logP_{RL}(s|a)$$

The main difference between Same Response Penalty (SRP) and Ease of Answering (EA) is we penalize based on the log probability of the RL model instead of Seq2Seq model. Our intuition is as the training progresses and the RL model avoids repetition, this reward penalty will decrease.

3.7.3 Conversation History (CH) and Conversational Direction (CD)

One way for the dialogue agent to optimize all our reward functions would be to respond with grammatically correct, semantically different yet still generic responses as long as responses are highly probable responses that do not repeat even if they have very little to do with a cohesive on topic conversation. There is not a clear reward function that captures the appropriateness of a response in the context of the conversation history, but such a function would be ideal. For example, if we are having a conversation about "coffee", it would be appropriate to responses to center around aroma, beans, coffee shops, the morning, mugs, caffeine etc. It would also be appropriate for the conversation to drift into the topic of methods of "staying awake" but it should do so incrementally and slowly with coherent transitions. We propose a reward function that encourages an intermediate cosine similarity between consecutive responses that we call Conversational Direction (CD).

$$CD = IF - K(C - \frac{h1 \cdot h2}{||h1||||h2||})^2$$

C is a hyper parameter $\in$ [0.5,0.9], K is a hyper parameter $\in$ [10,100] and $\frac{h1 \cdot h2}{||h1||||h2||}$ is the cosine distance between the hidden state encoding of the conversation history and the current response. To give some intuition to what cosine similarity of encodings is measuring, the cosine similarity between "hi how are you?" and "how are you?" is 0.94, whereas the cosine similarity between "hi how are you?" and "i don't what to" is 0.45.

3.7.4 Adding a baseline model

It is often the case that the vanilla policy gradient techniques for reinforcement learning have high variance and this problem can be mitigated by subtracting a baseline which is a function of the state. In this setting, the loss function is given by:

$$L(W) = E_{policy}[\sum_{t=1}^{T} - (R(a_t|p,q) - b([p,q]))\, log(P(a_t|p,q))]$$

In the above equation, b is a baseline model which takes in a sequence of the previous and current dialogue concatenated together and produces an estimate of the reward for that state. After each episode, the baseline is refit to minimize the squared difference between the predicted value and the empirical return received by the agent for each timestep for the conversation, ie. $(b([p,q]) - R_t)^2$ where R_t is simply the return after timestep t

The baseline model used in this experiment was an LSTM encoder where the final hidden state is fed to a fully connected layer to produce a single number. Empirically, it was found that adding a baseline helped to make the convergence of the average reward more stable and this could potentially be improved further by adjusting the parameters of the baseline network.

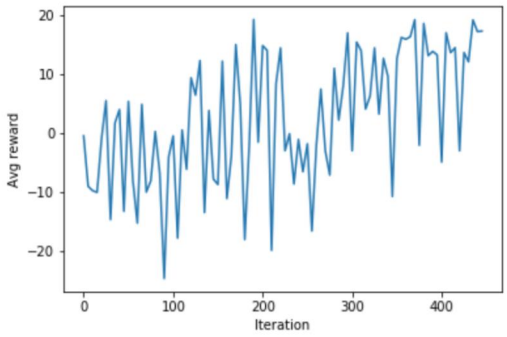

Figure 2: Plot of avg. reward for each episode using a baseline model

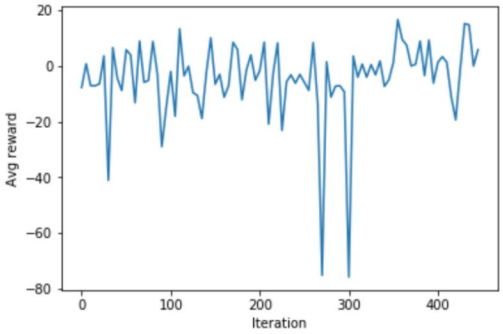

Figure 3: Plot of avg. reward without a baseline model

4 Evaluation methodology

For quantitative evaluation, we will use similar metrics as [2], the length and diversity of the dialogue, where we expect a good dialogue system can produce diverse topics or responses at a reasonable dialogue length. The length of the dialogue is measured from the start of the dialogue until the first nonconstructive response (e.g. repetition or dull response) is seen, while the diversity is measured from the scaled number of unique unigram and bigrams in the dialogue. The test questions used to initialize the conversation were taken from an independent source, a website consisting of 200 interesting questions to get to know someone. https://conversationstartersworld.com/questions-to-get-to-know-someone/.

5 Results

We found our best results after experimentation with the coefficients for each reward to be IF = 5.0, EOA = 1.0, SC = 1.4.

In Figure 4, based on the length of the conversation and n-gram diversity, there is a significant improvement in the diversity of language used in the simulated conversations based on the unigram and bigram ratios. We also observe significantly longer conversation length for the RL model.

Qualitatively, as we can see from the examples in the next section, the Seq2Seq model tends to repeat itself after the second timestep in the conversation. On the other hand, the RL model shows a different behavior depending on the reward we implement. For Ease of Answering (EA) reward only, we observe that even though the model manages to avoid generating dull responses from our training set, it instead generates a different variant of dull responses, such as ”I don’t know what happened” and ”I don’t know what happened to me?”. For Information Flow (IF) reward, the model outputs a more fluent conversation, exchanging different responses with minimal repetition. Finally, with the three rewards combines, we observe a diverse and more fluent conversation from the RL model than the baseline Seq2Seq model. In addition, the example dialogue shows an interesting exchange between the two RL agents.

For the additional strategies, in Figure 5, we can see that incorporating decaying ϵ-greedy learning can improve the length of the conversation at the cost of repetition. This repetition might be caused by the greedy action, where the RL model tends to imitate the Seq2Seq repetition as the best action. On the other hand, RL model with Same Response Penalty (SRP) has better unigram and bigram diversity. As we can see in the example output, the model generates more diverse responses as it avoids repeating previous remarks.

Using the conversation history and rewarding an intermediate cosine distance added a significant amount of variance to the reinforcement learning of the model as well as additional hyper-parameters to tune. It is not clear whether a significant improvement was gained from this particular formulation of the reward function.

5.1 Baseline Seq2Seq Model

Example output:

```
B: tell me something cool
A: and sad ! <EOS>
B: what do you mean ? <EOS>
A: thats thats why <EOS>
B: what are you doing ? <EOS>
A: thats thats why <EOS>
B: what are you doing ? <EOS>
A: thats thats why <EOS>
```

Reinforcement Learning - Ease of Answering only

```
B: who are you?
A: i dont know what happened ? <EOS>
B: what are you going to do ? <EOS>
A: wha about him ? <EOS>
B: i dont know what happened to me ? <EOS>
```

Reinforcement Learning - Information Flow only

```
B: tell me something cool
A: you can do it <EOS>
B: i guess <EOS>
A: what do you mean ? <EOS>
B: what are you doing ? <EOS>
A: i saw it up ! <EOS>
B: what do you mean ? <EOS>
A: wha about him ? <EOS>
```

Reinforcement Learning - Ease of Answering, Information Flow, Semantic Coherence

```
B: my name is carson
A: hi how are you sam ? <EOS>
B: hi how are you ? <EOS>
A: how am i ? i am having a
existential crisis
B: i think we should go back back
back back back
A: i didnt have to go back <EOS>
B: aw what are you going to do ? <EOS>
A: i dont think about it <EOS>
B: aw what are you talking about ? <EOS>
A: im not going to go back back
back back back
B: aw where are we going to go back
back back
A: im thinking about it <EOS>
B: learning how many times
are you sure ? <EOS>
```

Reinforcement Learning - Decaying ϵ-greedy

```
B: my name is carson
A: oh how are you sam ? <EOS>
B: i want to see you <EOS>
A: i think its a minute <EOS>
B: but what do you mean ? <EOS>
```

Metrics	Seq2Seq	RL
Avg number of turns	1.3	5.79
Avg unigram diversity ratio	0.278	0.498
Avg bigram diversity ratio	0.325	0.632
Dull word count ratio	5.85	0.0048

Figure 4: **Results.** Baseline result for Seq2Seq and Reinforcement Learning model from the generated dialogue

Metrics	ϵ-greedy	SRP	CHCD
Avg number of turns	6	5.976	5.79
Avg unigram diversity ratio	0.472	0.628	0.49
Avg bigram diversity ratio	0.559	0.768	0.63
Dull word count ratio	0.0	0.0	0.005

Figure 5: **Additional Strategy Results.** Result for additional experiment models from the generated dialogue

```
A: i want to see you <EOS>
B: i think its a minute <EOS>
```

Reinforcement Learning - Same Response Penalty

```
B: my name is carson
A: jenny how are you sam ? <EOS>
B: jenny baby its just the only
way we have to
A: oh this is not the only way i want to
B: she said the only way you know ? <EOS>
A: ask me the time i saw it <EOS>
```

Reinforcement Learning - Conversation History (CH) and Conversational Direction (CD)

```
B: the weather is nice today <EOS>
A: i think so im not sure about it <EOS>
B: i think i ll be very happy
to discuss it
A: i know i ll be very happy
to discuss it
B: i am very disappointed for it <EOS>
A: i think i ll be very happy
to discuss it
B: i think i ll be very happy
to discuss it
```

6 Discussion

Neural sequence to sequence models have demonstrated promising results for language translation. Indeed, within the first few epochs of training, a seq2seq model trained on dialogue learns to speak in proper English grammar very quickly. However the model tends to learn to respond in ”safe”, highly probable responses that are dull and lead to un-interesting conversation. We explore in this study the effect of different reward functions that can be used in a deep reinforcement learning setting to teach a pre-trained sequence to sequence model for the purpose of diverse dialogue generation. Several teams have tackled this problem with different approaches including adversarial strategies [8] and have been able to improve standard metrics of performance that correlate with human annotation [9]. However it remains a difficult task to capture and reward for conversations that take into account long term context and perform successfully a conversational goal, which is the point of most conversations. These would be interesting areas of for future research.

7 Code repository

https://github.com/clam004/RL-Chat-pytorch

References

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[2] 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.

[3] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

[4] Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87. Association for Computational Linguistics, 2011.

[5] Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv preprint arXiv:1606.00372*, 2016.

[6] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[7] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

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

[9] Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. *arXiv preprint arXiv:1701.03079*, 2017.