Dancing Seq2Seq: An Adversarial Approach to Transformer Generative Dialog

Alan Salimov [*1]

Abstract

Generative dialogue systems use RNNs, attention mechanisms, and adversarial training to create coherent dialogue on both closed and open topic corpora. While previous state of the art approaches used recurrent encoder-decoder(HRED) approaches (Serban et al., 2015), the transformer approach (Vaswani et al., 2017) has shown to provider faster and better results across NLP applications. By using stacked layers of attention instead of RNNs, transformers have shown that a language model can be learned using solely attention. However, all of these approaches suffer from a lack of diverse responses. HRED trained adversarially (Olabiyi et al., 2018) has shown to outperform vanilla HRED; in this paper, we propose an adversarially trained generative dialog system, mirroring the utterance level discriminator proposed in (Olabiyi et al., 2018). We generate dialog using a multi-gpu adapted version of the transformer seq2seq system proposed in (Adeniji et al., 2019). We train this using the Ubuntu Help Forum Dialog Corpus, a closed-topic corpus. Our two goals are to incorporate a much larger dataset and to derive improvements mirroring those in (Olabiyi et al., 2018).

1. Introduction and Related Work

Generative dialog, specifically multi-turn chit-chat, is a developing field in NLP. Current approaches leverage seq2seq models with recurrent structures (Serban et al., 2015), adversarially trained variations (Olabiyi et al., 2018), persona- and identity-based approaches (Olabiyi et al., 2019), and attention based models (Vaswani et al., 2017; Adeniji et al., 2019). The purpose of these variations is remember important information throughout a conversation, avoid both nonsense and generic responses, and provide an understanding of syntax, semantics, and consequences in a generative dialog model.

[1]Stanford University, Capital One. Correspondence to: Alan Salimov <alan@alansalimov.com>.

CS230, Fall 2019

While the original attention model in NLP (Vaswani et al., 2017) was primarily a question-answer based model, (Adeniji et al., 2019) synthesized the stacked layers of self-attention models with session-level recurrent memory, placing this between the encoder and decoder. As mentioned in their future work section, their model is bottlenecked by a primitive argumentation dataset. A more extensive corpus, like the Ubuntu Help Forum, presents issues with reasonable training times and model complexity. In addition, the issue of diversity of responses mentioned in hred-gan (Olabiyi et al., 2018) remain; the authors of the Transformer seq2seq model (Adeniji et al., 2019) mention the same issues present in the original HRED paper (Serban et al., 2015).

The work presented in this paper involves significant architectural shifts to allow for reasonable training times while maintaining appropriate model complexity by using both parallel model and loss function criteria (Zhang et al., 2018).

To address the diversity issue, we use the GAN approach given in hred-gan to reduce over-generalization ("I don't know") responses by removing the maximum-likelihood criteria and instead simply answering: is this response a human or computer generated response? While adversarial approaches have been used in the transformer architecture for machine translation (Wang et al., 2019), as in (Olabiyi et al., 2018), training a generative dialog model adversarially using the transformer has not been done.

2. Model Architecture

2.1. Generator

The dialogue generator is a slightly-modified seq2seq attention model (Adeniji et al., 2019), which is an implementation that modifies the atttention-based single turn dialogue system from (Vaswani et al., 2017), also known as the Transformer.

For each step in a given sequence, the generator uses both the positional word-level embeddings used in (Vaswani et al., 2017), concatenated with GloVE embeddings learned during training. These are then fed through an encoder, which is a stack of multi-headed self-attention layers. The output is maxpooled and fed into an LSTM to create a globally-aware query, which is attended to using dot-product attention. Finally, using an equivalent stack of multi-headed

self-attention layers for the decoder, predictions for that step in the sequence are generated. To increase training speed, we calculate the cross-entropy loss and update gradients for each step in the sequence, which differs from the original generator implementation.

2.2. Discriminator

The discriminator is adapted from the utterance level discriminator given in hred-gan, described in 1. To allow gradients to flow back into the generator, rather than feeding the sequence directly into the discriminator, we feed the embedding output of the decoder directly into an LSTM, which is then passed into a linear layer and sigmoid with an output size of 1.

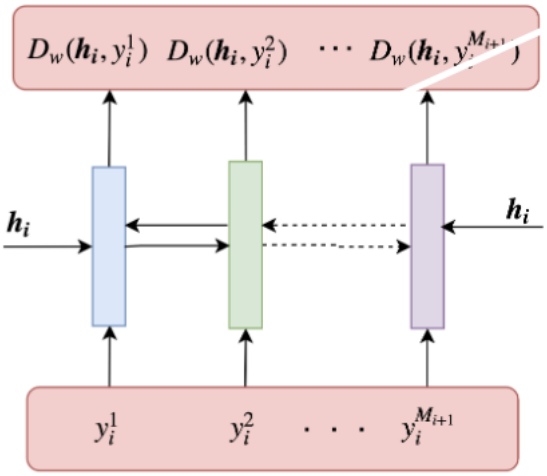

Figure 1. The utterance level discriminator used in hred-gan.

3. Training

3.1. Adversarial Training

To train the generator, we calculate the binary cross-entropy loss between the generator output, fed through the discriminator, and and a tensor of all 1s for each step, updating gradients through the discriminator and the generator.

We calculate the binary cross-entropy loss between the generator-¿discriminator predictions and a tensor of all 0s to train the discriminator.

3.2. Multi-GPU Training

CUDA memory constraints in (Adeniji et al., 2019) limited batch size to 4, and training set size to 11800. The parallelism we introduced allowed for a batch size of 1800, 64 data loader workers, and an increase of training set size to 1,645,200, with a vocabulary size of 50000. The generator-only version of training has 10 million parameters, while the full adversarial monty uses roughly 16 million parameters.

To facilitate somewhat reasonable training times, we use the implementation of both DataParallelModel and DataParallelCriterion implemented in (Zhang et al., 2018). This allows each GPU to calculate the losses for that particular batch and quickly reduce them. This approach reduces per-epoch training time from more than 24 hours to 3 hours over the full training set. As in the original (Adeniji et al., 2019), we used a scheduled Adam optimizer with 4000 warm-up steps.

Hyperparameters were halved to accommodate the demands of the new training set. We used an embedding size of 128 (smaller than the standard 300), inner hidden dimension of 128 for all LSTMs, and key/value dimensions of 32.

4. Dataset

Ubuntu Dialogue Corpus, (UDC) dataset (Olabiyi et al., 2018). This dataset was extracted from the Ubuntu Relay Chat Channel. The dataset is very large compared to the Internet Argument Corpus, and is closed topic; all dialogs are related specifically to Ubuntu Forum topics. The UDC contains about 1.85 million conversations with an average of 5 utterances per conversation, with a maximum per-utterance sequence length of 40. The maximum number of utterances is 25.

Preprocessing is done similarly to the approach presented in (Adeniji et al., 2019). Some minor adaptions needed to be done to accommodate the new dataset, such as fixing the vocabulary size.

5. Results Discussion

Unfortunately, due to the difficulty in implementing the parallel processing and issues with the training instance, we were unable to derive perplexity and inference examples from the new adversarial architecture or the original (Adeniji et al., 2019) model with the new dataset. The primary upshot of this work was to adapt the Transformer based seq2seq architecture to work with the much larger than those presented in the original paper (Adeniji et al., 2019). This was successful, as discussed in the Adversarial Training section, and the conversion of the work presented in (Mei et al., 2016) using the state of the art DataParallelModel/Criterion is significant in its own right.

6. Conclusion and Future Work

While this is just the first (unsuccessful) step into adversarial trained transformer-based generative dialog models, we do finally have access to relativly quickly trained models (a little slower than the hred-gan on the same corpus.) Addi-

tonal optimizations can involve iteration on the DataParallelModel/Criterion to work specifically for this application; (Zhang et al., 2018) was primarily used for CNNs, and there is a possibility that it can be optimized specifically with the transformer architecture in mind.

Given that transformers are a fairly new development that still suffer from the diversity issue, future work in the generative dialog space can follow the rough path the HRED track did. The Latent Variable HRED model (Serban et al., 2016) injected random noise in an effort to force the model to generate diverse responses; similar noise can be injected into the global memory.

Optimized implementation of the adversarial effort should lead to better results with respect to the diversity problem. (Wang et al., 2019) did show that adversarially trained transfomers for machine translation generate a more diverse embedding space. The insights gleamed from (Wang et al., 2019), (Olabiyi et al., 2018), and (Adeniji et al., 2019) do show that improvements for Transformer based generative dialog approaches can take the same path as did the iterations on HRED.

References

Adeniji, A., Lee, N., and Liu, V. Sequence-to-sequence generative argumentative dialogue systems with self-attention. 2019. URL https://web.stanford.edu/class/cs224n/reports/custom/15844523.pdf.

Mei, H., Bansal, M., and Walter, M. R. Coherent dialogue with attention-based language models. *CoRR*, abs/1611.06997, 2016. URL http://arxiv.org/abs/1611.06997.

Olabiyi, O., Salimov, A., Khazane, A., and Mueller, E. T. Multi-turn dialogue response generation in an adversarial learning framework. *CoRR*, abs/1805.11752, 2018. URL http://arxiv.org/abs/1805.11752.

Olabiyi, O., Khazane, A., Salimov, A., and Mueller, E. T. An adversarial learning framework for A persona-based multi-turn dialogue model. *CoRR*, abs/1905.01992, 2019. URL http://arxiv.org/abs/1905.01992.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C., and Pineau, J. Hierarchical neural network generative models for movie dialogues. *CoRR*, abs/1507.04808, 2015. URL http://arxiv.org/abs/1507.04808.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A. C., and Bengio, Y. A hierarchical latent variable encoder-decoder model for generating dialogues. *CoRR*, abs/1605.06069, 2016. URL http://arxiv.org/abs/1605.06069.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Wang, D., Gong, C., and Liu, Q. Improving neural language modeling via adversarial training. *CoRR*, abs/1906.03805, 2019. URL http://arxiv.org/abs/1906.03805.

Zhang, H., Dana, K. J., Shi, J., Zhang, Z., Wang, X., Tyagi, A., and Agrawal, A. Context encoding for semantic segmentation. *CoRR*, abs/1803.08904, 2018. URL http://arxiv.org/abs/1803.08904.