10-707 Final Report

Eric Zhu ericzhu@andrew.cmu.edu
Alex Xiao axiao@andrew.cmu.edu
Darshan Patil dapatil@andrew.cmu.edu

1 Introduction

There has been a recent surge of interest in generative neural encoder-decoder models for dialogue due to their improved performance on unstructured, open-domain generation over previous dialogue models. They have increasingly been able to generate more diverse and coherent responses and are also able to learn from data without the need of extensive domain knowledge or hand-crafting.

These improvements have been largely driven by extensions to neural architectures for sequential data that explicitly model context, beginning with Serban et al (2015) [13]. These models approach dialogue similar to neural machine translation by using sequence-to-sequence models among utterances within a dialogue. These models, however, did not produce satisfactorily meaningful or diverse responses, potentially due to the "shallow generation process" in which a model's sole source of variation is modeled at the step level. This leads models to ignore context in favor of improved predictions at a local level.

Kingma and Welling (2013) [7] introduce the variational autoencoder (VAE), a generative model that has seen strong success in image generation through the use of latent variables. VHRED (Serban et al., 2016) [12] capitalizes on this generative framework to model discourse-level variability, which alleviates some of the earlier problems with uninteresting responses produced by the models. The authors found that maximizing log-likelihood (i.e. minimizing perplexity) is not a sufficient training objective for dialogue models, as simpler models performed better on that objective but worse in human evaluations.

Neural models for sequential data that use variational methods are notoriously hard to optimize, due to the tendency for models to ignore the latent variable in favor of short-term gains. Many methods have been proposed to alleviate this issue, including Kullback–Leibler (KL) divergence cost annealing [2], bag-of-word (BOW) loss, and weakening the decoder.

This project aims to improve upon the stability of VHRED training, as well as the overall performance of VHRED for dialogue generation. We introduce a new method for KL annealing, ReLU annealing, to address some VHRED-specific optimization challenges. (See Section 3.3.) We incorporate an auxiliary loss term, bag-of-word (BOW) loss [15] to prevent the collapse of the VHRED latent variables. (Refer to Section 3.2.) Finally, we incorporate encoder-decoder attention [9] to improve upon low-context scenarios. (See Section 3.1.) We also train on a novel open-domain corpus, the Cornell Movies Dialogues Corpus [5], whereas previous work on HRED and VHRED has trained on either domain-limited or non-real-world dialogues.

2 Related Work

Significant progress has been made in neural dialogue generation in the past few years. We give an overview of the progress most relevant to our project.

10707 Class Project (2017).

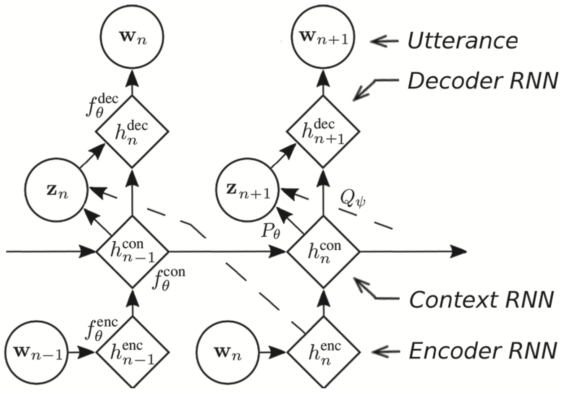

Figure 1: Latent Variable Hierarchical Recurrent Encoder-Decoder (VHRED) [12]

2.1 Encoder-Decoder Architecture

The goal of dialogue generation is to output an appropriate response when given the previous conversation. This can be modeled as sequence to sequence problem, where given the previous sentence $x = (x_1, x_2, ..., x_T)$ the model should output the response $y = (y_1, y_2, ..., y_{T'})$. Learning a sequence to sequence relationship can be done with an encoder and decoder architecture, as done in Cho et al (2014) [3]. The encoder and decoder architecture consists of two recurrent neural networks, which jointly models $p(y|x)$. The first RNN (the encoder) models $p(x_1, x_2, ..., x_T)$ by using the last RNN cell's state c as a summary of x. c is then passed to the decoder RNN, which models $p(y_j|c, y_1, ..., y_{j-1})$ at each time step to predict the response y.

2.2 Hierarchical Recurrent Encoder-Decoder (HRED)

Serban et. al [13] extend the encoder-decoder architecture to better model the dialogue generation setting. HRED consist of three components: an encoder RNN, a context RNN, and a decoder RNN. Instead of just taking into account the previous utterance when generating the response, HRED uses the entire previous conversation as context. Given a list of previous utterances $x^1, x^2, \ldots, x^n$ and target response y, HRED models $p(y|x^1, x^2, ..., x^n)$ instead of just $p(y|x^n)$. It does this by encoding each x^i with the encoder RNN and using the encoder RNN's ouptut i as input i to the context RNN. The final state c of the context RNN then represents $(x^1, x^2, ..., x^n)$. c is passed to the decoder RNN, which models $p(y_j|c, y_1, ..., y_{j-1})$ at each time step to predict the response y. The context RNN's hidden state is only updated with each utterance instead of each word, allowing it model higher level conversational features like conversation topic.

2.3 Latent Variable Hierarchical Recurrent Encoder-Decoder (VHRED)

Since HRED's [13] only source of variation is at the word output level, it tends to favor capturing local over global structure while training. As a result, its dialogue favors short term goals, leading it to ignore conversation context and generate more generic responses.

VHRED [12] attempts to fix this issue by extending the idea of HRED by introducing latent variables to the context RNN. In addition to the context hidden state c, VHRED also samples latent variables z from a prior distribution $N(\mu_{prior}, \Sigma_{prior})$, where $\mu_{prior} = f_\theta(c)$ and $\Sigma_{prior} = g_\theta(c)$. z is then concatenated with c and is passed into the decoder RNN. (See Figure 1.) Similar to VAEs [7], at training time VHRED infers the approximate posterior distribution of z with the ground truth response $Q(z|y)$, and at test time samples z from the prior. By introducing stochasticity at the sentence level, VHRED is better able to output diverse responses and model long contexts.

3 Method

We aim to improve upon the modeling capabilities of VHRED [12] in three ways: (1) by adding attention, (2) by adding BOW loss, and (3) using ReLU annealing.

3.1 Attention

We propose adding Luong attention [9] to our model by attending over the encoder RNN outputs and concatenating the resulting attentional vectors to the inputs to the decoder RNN.

The VHRED model outperforms the other baselines when given a long context, but does not perform as well as a simple LSTM model with a short context [12]. Furthermore, the decoder RNN only gets the context RNN output and the latent variables as input, which are also generated by the context RNN. By attending on the previous utterance in a dialogue, we give the decoder more information about the short term input given to the model. This allows us to explicitly model local context separately from long-term dependencies among utterances which gives the model the ability to produce a coherent response, even when it has less context.

3.2 Bag-of-Word (BOW) Loss

We include an auxiliary bag-of-word (BOW) loss as introduced in [15]. Often while training, KL-loss collapsed to 0, which resulted in the latent variable, z, becoming meaningless in the generation process. Adding the BOW loss requires the decoder network to be able to predict the bag of words in the generated response using only z. This essentially forces z to encode meaningful information about the response, and thus discourages the posterior from collapsing into the prior distribution for z and KL-loss from collapsing.

3.3 Rectified Linear Unit (ReLU) Annealing

Existing literature uses linear or sigmoid annealing of the KL cost term during training to prevent collapse of the KL cost term. We observe that even with a seemingly-negligible KL cost term (10^{-5}) at the onset of training, the KL loss is immediately driven to zero. We hypothesize that this is due to the random initialization of weights in the model. The prior and posterior latent distributions are both randomly initialized and so the latent variable z provides no meaningful information to the decoder. Thus any signal through the KL cost term causes the two distributions to collapse together in order to drive that cost term to zero, interfering with the model focusing on the reconstruction cost term. We would like the model to first begin learning how to reconstruct the data, so that it can eventually coordinate between having a meaningful latent z with small KL cost term and a good reconstruction loss.

Using sigmoid annealing with a KL cost term below 10^{-5} at step 0 causes the KL cost term to have a steep spike at its halfway point. This unexpectedly causes the model to coordinate between the two cost terms. We observe empirically that the reconstruction term immediately spikes, which is undesirable for stable training and better coordination.

To address these problems, we introduce ReLU annealing to the model's KL cost term. This allows the decoder to first begin learning before coordinating with the latent variables.

4 Dataset

Serban et al. [12] trained their model with the Ubuntu dataset [8] and the Twitter Dataset [11]. The Ubuntu dataset is very technical and is not a good representation for open-domain dialogue. The Twitter dataset is more representative of real world language, but the conversation structure resembles more of an email format rather than natural conversation.

To fix the above issues, we train our models using the **Cornell Movie Dialogues Corpus** [5]. This dataset contains 83,097 dialogues with 220,579 exchanges between 10,292 pairs of movie characters from 617 movies. Due to its wide variety of general, everyday topics, the Cornell Movie Dialogues Corpus can be used to train a general purpose conversation agent. To remove unnecessarily long

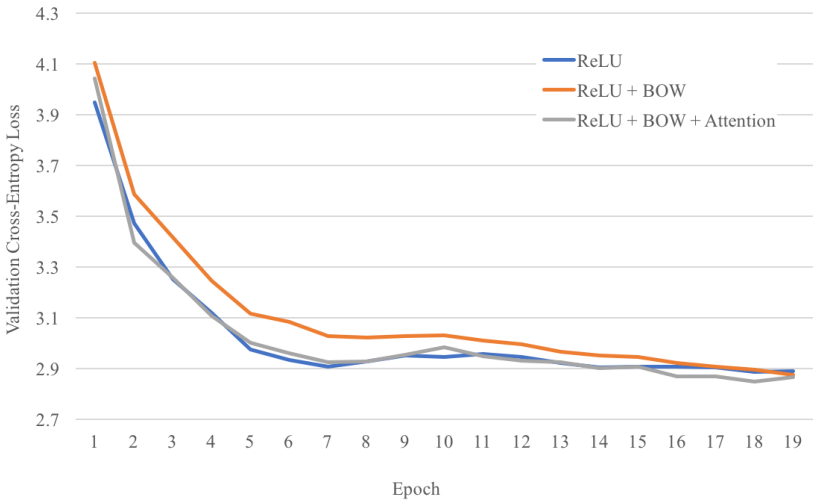

Figure 2: Reconstruction loss for models using ReLU annealing

Model	**Recon. Loss**	**KL Loss**	**BOW loss**	**Total Loss**
HRED	3.540	—	—	**3.540**
VHRED + linear anneal*	3.544	5.89 e-3	—	3.550
VHRED + sigmoid anneal*	3.582	6.061 e-3	—	3.588
VHRED + ReLU anneal	2.893	**4.292**	—	7.185
VHRED + BOW + no anneal + attention*	**1.085**	1083.059	**3.681**	1084.144
VHRED + BOW + ReLU anneal	2.876	4.843	4.093	7.719
VHRED + BOW + ReLU anneal+ attention	2.865	5.151	3.858	8.016

Table 1: Validation performance of each model on automatic measures. (Starred models are those where KL loss collapsed or exploded.) "Total loss" for models using variational methods refers to the variational lower bound on the loss.

dialogues that would be unhelpful for conversation, we further process the dataset by filtering out any dialogues containing utterances with more than 20 words. We then transformed the dialogues into 42,189 triples (x_1, x_2, x_3), where x_1, x_2, x_3 are adjacent utterances from a dialogue in the corpus. The models are then trained on these triples by trying to predict x_2 given x_1 and x_3 given x_2 and x_1. We use triples of dialogue utterances due to memory constraints.

5 Experiments and Results

We implemented both HRED [13] and VHRED [12] from scratch using Tensorflow. We trained with the following hyperparameters. We use an embedding size of 300 using embeddings pretrained using Word2Vec on the Google News corpus [10]. All latent variables have dimensionality of 100. We clip gradients to 2.5. We use encoder and decoder hidden state size of 512 and context RNN hidden state size of 1024. All RNNs are instantiated as gated recurrent units (GRU) [4]. We tried using bidirectional RNNs for the encoder with no success, so we did not include results for those models. All models were built using Tensorflow 1.4 [1].

We trained a baseline HRED model, followed by many VHRED models with various extensions, as shown in Table 1. Our sigmoid annealing function is $\sigma(20 + (-40/9000)x)$ where x is the step. We use a learning rate of $8 \cdot 10^{-3}$. The ReLU cost term coefficient begins increasing after 450 steps. All

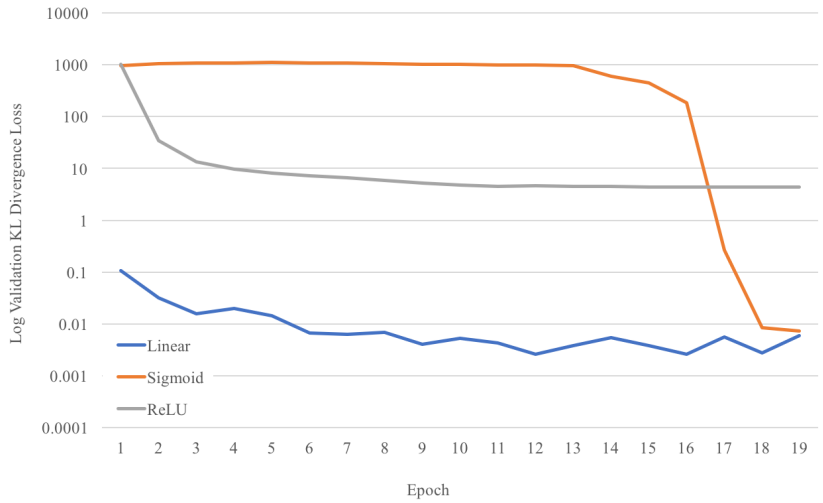

Figure 3: KL loss for models with different KL annealing

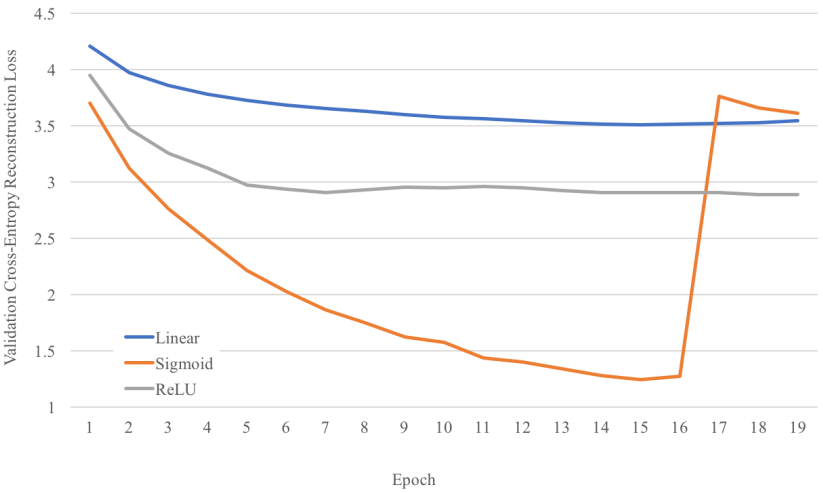

Figure 4: Reconstruction loss for models with different KL annealing

VHRED models that used non-sigmoid KL cost annealing annealed the KL cost coefficient to 0.1 at step 2000, though the reported cost is the full KL cost for use in the variational lower bound to the negative log likelihood. The convergence properties of the models are shown in Figures 2, 3, and 4.

To generate sentences from our models, we used beam search with a beam size of 15 and length penalty weight of 0.6 [14]. Generated outputs are shown in Table 2.

6 Discussion and Analysis

6.1 Annealing Methods

Figure 3 displays the KL loss convergence properties of the three different annealing methods. Linear annealing starts out too high and immediately causes KL divergence to collapse to 0, suggesting that the weight used for KL loss initially is too high. Sigmoid annealing fixes this problem, but causes the KL loss to crash at about epoch 16. We believe this is because KL loss has been ignored for too long,

Dialogue Context	Model Response
How are you?	**HRED**: I don't know. **VHRED linear**: I don't know. **VHRED ReLU**: I'd like to talk to you.
Watch out!	**HRED**: What is it? **VHRED linear**: Why? **VHRED ReLU**: Hey, what's the matter?
He said everyone was doing it, so I did it.	**HRED**: What happened? **VHRED linear**: What? **VHRED ReLU** : That's what I said.
I hate eating vegetables. → Wait, why?	**HRED**: <unk>? **VHRED linear**: I don't know . **VHRED ReLU** : Because what's the point of <unk> ?

Table 2: Dialogue generation results on Cornell dataset

which results in the gradients with respect to the KL loss suddenly dominating the gradient descent. This causes the KL loss to crash downwards, and the reconstruction loss to spike upwards, as shown in Figure 4.

ReLU annealing fixes both of the above problems. As shown in Figure 3, it displays a smooth downward trend and converges at about a loss of 4. It doesn't have the sudden spikes of sigmoid because the KL loss and reconstruction loss start coordinating from a much earlier time.

Furthermore, ReLU annealing doesn't collapse like linear annealing despite it being identical to linear annealing after the ReLU takes in positive inputs (450 mini-batches in our case). We theorize that this is because initial reconstruction error is so high due to the random weights in the RNN's such that the model will first attempt to optimize these weights instead of trying to make the latent variable z useful. As a result, the model is free to generate any z necessary to minimize KL, resulting in the linear annealing method causing KL loss to crash. Once the model gets past this initial stage, linear annealing allows it to gradually converge to a reasonable KL loss. Note that we could get a similar behavior by using linear annealing with an extremely small slope, but that would also make the model's KL divergence loss converge much more slowly than with ReLU annealing.

6.2 Attention and BOW Loss

Attending over the encoder states resulted in a slightly lower reconstruction loss than the rest of our non-collapsed models, but ended up with a higher KL loss than our model with no attention. Thus, we cannot conclusively say whether adding attention improves the performance of our model when dealing with short term contexts.

Using BOW loss resulted in higher KL losses, as expected. However, we found that ReLU loss is capable of preventing KL cost collapse on its own, as we see in Figure 3. Thus, adding the BOW loss became unnecessary, as its primary aim is to prevent the model from ignoring the latent term. Using BOW loss tended to result in models that generated worse responses, potentially because the auxiliary cost term prevented the model from better coordinating between the two other loss terms.

6.3 Chatting

Although the negative log likelihood and KL divergence provide insight into the performance of the different models, they were insufficient to measure model performance from the perspective of human judgment. When "chatting" with the model by generating responses to utterances, the VHRED model with ReLU annealing was substantially better in terms of response diversity and meaningfulness, despite having worse validation loss. This can be explained by the model successfully having a meaningful latent variable, allowing it to capture utterance and dialogue-level variability.

The VHRED model with linear annealing essentially collapsed to the HRED model because its KL loss collapsed to zero, so its chatting performance mirrored the HRED performance.

7 Future Plan

Since the VHRED authors mention that their model outperformed all other baselines on examples with longer dialogues, while it was outperformed by an LSTM model for shorter dialogues, we want to compare our model's performance on datasets with longer dialogues to see how it performs on longer context. Another future direction of research could be adding attention over the hierarchical states in addition to attention over the encoder states in order to allow the model to better handle long-term dependencies over the utterances.

The results above show that VHRED is capable of generating a coherent response given relevant context, but that response is not controllable and depends on the randomly sampled latent variables z. It is strongly preferred that dialogue agents have a maintain topic and have a sort of identity, making them more human-like in conversation. Future directions would explicitly model other states in the dialogue. Hu et al. [6] introduce a model that aims to generate plausible sentences conditioned on representation vectors which are endowed with designated semantic structures. Their approach combines VAEs with attribute discriminators and imposes explicit independency constraints on attribute controls, enabling disentangled latent code. One future direction of research could be combining the two models to move toward controllable dialogue generation.

8 Conclusion

In this paper, we proposed several potential improvements to the VHRED model. Adding attention over the encoder states resulted in the best reconstruction loss, but a slightly higher KL loss. We observed that adding BOW loss on its own did not help training. In conjunction with ReLU annealing, BOW loss simply became unnecessary and in fact resulted in worse results. We also found that adding ReLU annealing for the KL loss resulted in faster and much more stable training and the lowest KL loss for any model whose KL loss that did not collapse to 0. We also contribute a open-source implementation of HRED and VHRED with low memory utilization using dynamically-unrolled RNNs and the latest Tensorflow seq2seq libraries. We plan to release the code to the public.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning.

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

[3] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[5] 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 Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

[6] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Controllable text generation. *CoRR*, abs/1703.00955, 2017.

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

[8] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *CoRR*, abs/1506.08909, 2015.

[9] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[11] Alan Ritter, Colin Cherry, and William B. Dolan. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 583–593, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[12] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. *CoRR*, abs/1605.06069, 2016.

[13] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. *CoRR*, abs/1507.02221, 2015.

[14] Ziang Xie. Neural text generation: A practical guide. *arXiv preprint arXiv:1711.09534*, 2017.

[15] Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *CoRR*, abs/1703.10960, 2017.