Latent Variable Dialogue Models and their Diversity

Kris Cao and **Stephen Clark**
Computer Laboratory
University of Cambridge
United Kingdom
{kc391, sc609}@cam.ac.uk

Abstract

We present a dialogue generation model that directly captures the variability in possible responses to a given input, which reduces the 'boring output' issue of deterministic dialogue models. Experiments show that our model generates more diverse outputs than baseline models, and also generates more consistently acceptable output than sampling from a deterministic encoder-decoder model.

1 Introduction

The task of open-domain dialogue generation is an area of active development, with neural sequence-to-sequence models dominating the recently published literature (Shang et al., 2015; Vinyals and Le, 2015; Li et al., 2016b,a; Serban et al., 2016). Most previously published models train to minimise the negative log-likelihood of the training data, and then at generation time either perform beam search to find the output Y which maximises $P(Y|\text{input})$ (Shang et al., 2015; Vinyals and Le, 2015; Serban et al., 2016) (ML decoding), or sample from the resulting distribution (Serban et al., 2016).

A notorious issue with ML decoding is that this tends to generate short, boring responses to a wide range of inputs, such as *"I don't know"*. These responses are common in the training data, and can be replies to a wide range of inputs (Li et al., 2016a; Serban et al., 2016). In addition, shorter responses typically have higher likelihoods, and so wide beam sizes often result in very short responses (Tu et al., 2017; Belz, 2007). To resolve this problem, Li et al. (2016a) propose instead using maximum mutual information with a length boost as a decoding objective, and report more interesting generated responses.

Further, natural dialogue is not deterministic; for example, the replies to *"What's your name and where do you come from?"* will vary from person to person. Li et al. (2016b) have proposed learning representations of personas to account for inter-person variation, but there can be variation even among a single person's responses to certain questions.

Recently, Serban et al. (2017) have introduced latent variables to the dialogue modelling framework, to model the underlying distribution over possible responses directly. These models have the benefit that, at generation time, we can sample a response from the distribution by first sampling an assignment of the latent variables, and then decoding deterministically. In this way, we introduce stochasticity without resorting to sampling from the decoder, which can lead to incoherent output – see Table 1 for examples.

In this paper, we present a latent variable model for one-shot dialogue response, and investigate what kinds of diversity the latent variables capture. Our experiments show that our model has higher lexical as well as sentential diversity than baseline models. We also show that our model generates more acceptable diverse output than sampling from a deterministic decoder. We end by noting that all three methods proposed above to combat the 'maximum likelihood response' are ways of decreasing the probability of the generated output, and report some preliminary results for how response probability interacts with grammaticality and interestingness.

2 A Latent Variable Dialogue Model

2.1 Model Description

Our task is to model the true probability of a response Y given an input X. We denote our model distribution by $P(Y|X)$. We introduce a latent

Prompt	DIAL-LV	Sampling ($\tau = 1.0$)
the infection's spreading.	he's dead. nothing but no more. i know what you're doing	he was lee . leave it a way. there's something going on in the phone rickshaw and put the afloat continuously...
we only have fourteen hours to save the earth!	i've got no choice . and then there's something that's what it's like. well, we need the help of what's happening.	can i win? go peacekeeper go. go to say hello again in this conversation.
why don't i do some research and send you an email?	well, we're gonna have to make it. because i know how much it is. because it 's not right.	no disrespect team. excuse me. because i'm hotel.

Table 1: 3 random outputs for 3 random prompts from the dataset from our proposed model (DIAL-LV) and naively sampling from the decoder of a deterministic encoder-decoder.

variable z with a standard Gaussian prior – i.e. $P(z) = \mathcal{N}(0, I_n)$ – and factor $P(Y|X)$ as:

$$P(Y|X) = \int_z P(Y|z, X)P(z)dz \qquad (1)$$

To motivate this model, we point out that existing encoder-decoder models encode an input X as a single fixed representation. Hence, all of the possible replies to X must be stored within the decoder's probability distribution $P(Y|X)$, and during decoding it is hard to disentangle these possible replies.

However, our model contains a stochastic component z in the decoder $P(Y|z, X)$, and so by sampling different z and then performing ML decoding on $P(Y|z, X)$, we hope to tease apart the replies stored in the probability distribution $P(Y|X)$, without resorting to sampling from the decoder. This has the benefit that we use the decoder at generation time in a similar way to how we train it, making it more likely that the output of our model is grammatical and coherent. Further, as we do not marginalize out z when decoding, we no longer perform exact maximum likelihood search for a reply Y, and so we hope to avoid the boring reply problem.

At training time, we follow the variational autoencoder framework (Kingma and Welling, 2014; Kingma et al., 2014; Sohn et al., 2015; Miao et al., 2016) , and approximate the posterior $P(z|X, Y)$ with a proposal distribution $Q(z|X, Y)$, which in our case is a diagonal Gaussian whose parameters depend on X and Y. We thus have the following evidence lower bound (ELBO) for the loglikelihood of the data:

$$\log P(Y|X) \geq -\mathcal{KL}(Q(z|X, Y)||P(z))$$
$$+ \mathbb{E}_{z \sim Q} \log P(Y|z, X) \quad (2)$$

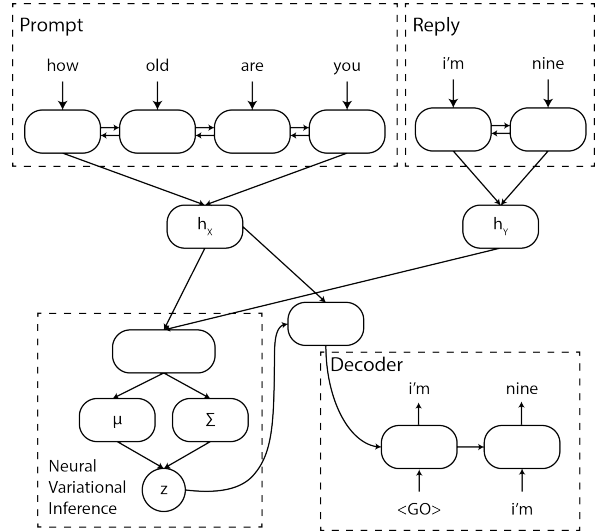Note that this loss decomposes into two parts: the KL divergence between the approximate pos-

Figure 1: A schematic of how our model is implemented. Please see the text for full details.

terior and the prior, and the cross-entropy loss between the model distribution and the data distribution. If the model can encode useful information into z, then the KL divergence term will be non-zero (Bowman et al., 2016). As our model decoder is given a deterministic representation of X already, z will then encode information about the variation in replies to X.

2.2 Model Implementation

Given an input sentence X and a response Y, we run two separate bidirectional RNNs over their word embeddings $\mathbf{x}_i$ and $\mathbf{y}_i$. We concatenate the final states of each and pass them through a single nonlinear layer to obtain our representations $h_{\mathbf{x}}$ and $h_{\mathbf{y}}$ of X and Y. We use GRUs (Cho et al., 2014) as our RNN cell as a compromise between expressive power and computational cost.

We calculate the mean and variance of Q as:

$$\mu = W_\mu[h_{\mathbf{x}} \, h_{\mathbf{y}}] + b_\mu$$
$$\log(\Sigma) = diag(W_\Sigma[h_{\mathbf{x}} \, h_{\mathbf{y}}] + b_\Sigma) \qquad (3)$$

where $[a\ b]$ denotes the concatenation of a and b, and $diag$ denotes inserting along the diagonal of a matrix.

We take a single sample z from Q using the reparametrization trick (Kingma and Welling, 2014), concatenate $h_{\mathbf{x}}$ and z, and initialize the hidden state of the decoder GRU with $[h_{\mathbf{x}}\ z]$. We then train the decoder GRU to minimize the negative log-likelihood of the response Y.

While training this model, we noted the same difficulties as Bowman et al. (2016) – as RNNs are powerful density estimators, the model will prefer to ignore the latent variables and instead optimize the data reconstruction term of the ELBO, while forcing the KL term to 0. We overcome this using similar techniques by gradually annealing the KL term weight over the course of model training and using word dropout in the decoder with a drop rate of 0.5.

3 Experiments

We compare our model, DIAL-LV, to three baselines. The first is an encoder-decoder dialogue model with ML decoding (DIAL-MLE). The second baseline model implements the anti-LM decoder of Li et al. (2016a) (DIAL-MMI) on top of the encoder-decoder, with no length normalization. For these models, we use beam search with a width of 2 to find the sentence Y which maximises the decoding objective (either ML or MMI).

The final baseline uses the encoder-decoder model, but instead samples from the decoder to find Y (DIAL-SAMP). We found that naively sampling from the decoder resulted in meaningless jumbles of words. To solve this, we introduced a temperature parameter $\tau \in (0, 1]$, which scales the probability of each word of the decoder as $p_w \mapsto p_w^{1/\tau}$. This parameter serves to sharpen the word distribution of the decoder. We found $\tau = 0.35$ to be a reasonable balance between preserving stochasticity while also improving the coherence of the generated output.

We used the OpenSubtitles dataset of movie subtitles to train our models (Tiedemann, 2012). We took a random sample of 100K files from the full dataset to train our models on, and then pruned this of repeated files to leave roughly 95K files and capped sentence length to 50. The total size of the resulting corpus was around 731M tokens. Please see the supplementary material for model hyperparameters and training details.

Model	Zipf parameter	NLL	Unique %
DIAL-LV	1.39	15.54	76
DIAL-MLE	1.43	12.15	35
DIAL-MMI	1.60	15.12	62
DIAL-SAMP	1.53	16.66	78

Table 2: Some statistics pertaining to the responses generated by the models.

As seeds for our replies, we used list of 50 prompts: 150 lines from the OpenSubtitles dataset outside of our training set which we judged to make sense as independent sentences and 50 questions chosen from a list of suggested conversation starters[1].

3.1 Reply statistics

Previous work (e.g. Li et al. (2016a)) used type-token ratio (TTR) to measure the diversity of the generated output. However, as language follows a Zipf distribution, TTR is affected by the length of the generated replies (Mitchell, 2015). Hence, we use the estimated parameter of a Zipf distribution fitted to our replies as a proxy for the lexical diversity of generated output, with more diverse output having smaller scores. As ML decoding is known to give the same few replies repeatedly, we also report the percentage of unique replies, as a coarser measure of sentential diversity compared to lexical diversity. Further, we give the negative log-likelihood (NLL) as predicted by the deterministic encoder-decoder model, to see what regions of the probability space the replies occupy. We present these statistics in Table 2.

We note that DIAL-LV generates more diverse replies than the other deterministic models, measured in terms of percentage of unique responses. Interestingly, the lexical diversity of DIAL-LV is almost identical to DIAL-MLE, suggesting that the latent variables help DIAL-LV avoid the boring output problem and generate more diverse outputs. We note that DIAL-LV even rivals DIAL-SAMP in terms of sentential diversity, and beats DIAL-SAMP in terms of lexical diversity. This could be because DIAL-SAMP chooses words greedily, and so is biased towards choosing high-probability words at each timestep. This suggests that maintaining a beam of hypotheses while sampling could help sampling-based methods escape

---

[1]Obtained from http://conversationstartersworld.com/250-conversation-starters/

Model	μ	σ	NLL	Zipf	Unique %
DIAL-LV	1.183	0.402	15.51	1.32	76.4
DIAL-SAMP	1.196	0.577	16.91	1.56	73.6

Table 3: Mean and std. dev. of average number of acceptable replies generated by each model.

Shell radius	Zipf parameter	NLL	Unique %
0	1.49	13.12	7
4	1.62	14.02	42.1
8	1.59	15.72	63.1
12	1.56	17.65	67.7
16	1.78	18.16	67.1

Table 4: Statistics of responses generated from the DIAL-LV model from different regions of the hidden state space.

the trap of having to make near-greedy local decisions.

3.2 Human acceptability judgments

We also tested whether DIAL-LV could generate a greater number of acceptable replies to a prompt than DIAL-SAMP. We randomly selected 50 prompts from our list of 200, and generated 5 replies at random to each one using both models. We then asked human annotators[2] to judge how many replies were appropriate replies, taking into account grammaticality, coherence and relevance. The results are shown in Table 3.

Interestingly, even though DIAL-LV has a lower NLL score, both models generate roughly the same number of acceptable replies. DIAL-LV also has less variance in the number of acceptable replies, suggesting that the outputs it generates are more consistent than responses from DIAL-SAMP. Finally, we note that DIAL-LV generates more diverse output than DIAL-SAMP in this scenario, even thought its replies are judged equally acceptable, suggesting that it is managing to produce a wide range of coherent, fluent and appropriate output.

3.3 Sampling from the latent variable space

We next explored the effect of sampling from different regions of the latent space. For each prompt in the test set, we took 5 uniform samples from shells of radius 0 (which collapses to determinis-

tic decoding), 4, 8, 12 and 16 in the latent space[3] by sampling from $P(z) = \mathcal{N}(0, I)$ and then scaling the sample z by the appropriate amount. We then generated a response to the prompt using each value of z, and measured some statistics of the replies. The results are shown in Table 4.

As expected, samples with small radius show less diversity in terms of unique outputs. Further, we see a consistent trend that samples with greater radius have a higher NLL score, showing the influence of the prior in Eqn. 1. However, at the highest radius, we observe the highest NLLs, but also the lowest lexical diversities, suggesting that it manages to combine the words it produces in many different ways.

4 Discussion

Taken together, our experiments show that ML decoding does not seem to be the best objective for generating diverse dialogue, and so corroborates the inadequacy of perplexity as an evaluation metric for dialogue models (Liu et al., 2016). Indeed, all three models which show a diversity gain over the vanilla encoder-decoder with MLE decoding try to instead sample responses from a lower-probability region of the response space. However, if the response probability is too low, it runs the risk of being nonsensical. Hence, there appears to be a 'Goldilocks' region of the probability space, where the responses are interesting and coherent. Finding ways of concentrating model samples to this region is thus a potentially promising area of research for open-domain dialogue agents.

We also note that our proposed model can be combined with MMI decoding or temperature-based sampling to get the benefits of both worlds. While we did not do this in our experiments in order to isolate the impact of our model, doing so improves the diversity of our generated output even more.

5 Conclusion

In this paper, we present a latent variable model to generate responses to input utterances. We investigate the diversity of output generated from this model, and show that it improces both lexical and sentential diversity. It also generates more consistently acceptable output as judged by humans compared to sampling from a decoder.

References

Anja Belz. Probabilistic generation of weather forecast texts. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 164–171, Rochester, New York, April 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N/N07/N07-1021`.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/K16-1002`.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.

Franois Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *ICLR*, 2014.

Diederik P. Kingma, Danilo Jimenez Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *NIPS*, 2014.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016a. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N16-1014`.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany, August 2016b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1094`.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas, November 2016. Association for Computational Linguistics. URL `https://aclweb.org/anthology/D16-1230`.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. *ICML*, 2016.

David Mitchell. Type-token models: a comparative study. *Journal of Quantitative Linguistics*, 2015.

Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. *Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. *AAAI*, 2016.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, 2017.

Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Lin-*

guistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1577–1586, Beijing, China, July 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-1152`.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprints*, abs/1605.02688, 2016. URL `http://arxiv.org/abs/1605.02688`.

Jörg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7. URL `http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf`. ACL Anthology Identifier: L12-1246.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Yang Liu. Neural machine translation with reconstruction. In *AAAI*, 2017.

Orial Vinyals and Quoc V. Le. A neural conversation model. *ICML*, 2015.

Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012. URL `http://arxiv.org/abs/1212.5701`.

A   Model training information

We implemented all of our models using Keras (Chollet, 2015) running on Theano (Theano Development Team, 2016). As vocabulary, we took all words appearing at least 1000 times in the whole corpus. As this amounted to ∼30K words, we used a 2-level hierarchical approximation to the full softmax to speed up model training (Morin and Bengio, 2005), with random clustering. We trained all our models for 3 epochs using the Adadelta optimizer (Zeiler, 2012), with default values for the optimizer parameters.

We used 512 dimensional word embeddings and encoder hidden state sizes across all of our models. We used 64 latent dimensional latent variables, and so the decoder RNN for the DIAL-LV model had hidden state size 574. The decoder RNN for the DIAL-MLE model also had hidden state size 574, to keep the capacity of the decoder comparable across the two models. We used tanh non-linearities throughout our model. For training the vanilla encoder-decoder, we also used word dropout on the decoder input with a drop rate of 0.5 to prevent overfitting. Each epoch took roughly 4 days on a Titan Black.

For the MMI decoding, we used a LM penalty weight of 0.45 and applied this for the first 6 words.