A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues

Iulian Vlad Serban
University of Montreal
2920 chemin de la Tour,
Montréal, QC, Canada

Alessandro Sordoni*
Maluuba Inc
2000 Rue Peel,
Montréal, QC, Canada

Ryan Lowe
McGill University
3480 Rue University,
Montréal, QC, Canada

Laurent Charlin
HEC Montréal
3000 chemin de la Côte-Sainte-Catherine, Montréal, QC, Canada

Joelle Pineau
McGill University
3480 Rue University,
Montréal, QC, Canada

Aaron Courville and **Yoshua Bengio**
University of Montreal
2920 chemin de la Tour,
Montréal, QC, Canada

Abstract

Sequential data often possesses hierarchical structures with complex dependencies between sub-sequences, such as found between the utterances in a dialogue. To model these dependencies in a generative framework, we propose a neural network-based generative architecture, with stochastic latent variables that span a variable number of time steps. We apply the proposed model to the task of dialogue response generation and compare it with other recent neural-network architectures. We evaluate the model performance through a human evaluation study. The experiments demonstrate that our model improves upon recently proposed models and that the latent variables facilitate both the generation of meaningful, long and diverse responses and maintaining dialogue state.

Introduction

Recurrent neural networks (RNNs) have recently demonstrated excellent results on a number of machine learning problems involving the generation of sequential structured outputs (Goodfellow, Courville, and Bengio 2015), including dialogue (Vinyals and Le 2015; Sordoni et al. 2015b; Serban et al. 2016), language modelling (Graves 2012; Mikolov and others 2010) machine translation (Sutskever, Vinyals, and Le 2014; Cho and others 2014) and speech recognition (Hinton and others 2012). However, the underlying RNNs often follow a shallow (flat) generation process, where the model variability or stochasticity only occurs when an output (e.g. word) is sampled. Injecting all the variability at the output level is often limiting, because the model is forced to generate all high-level structure locally on a step-by-step basis (Boulanger-Lewandowski, Bengio, and Vincent 2012; Bayer and Osendorfer 2014; Chung et al. 2015; Denton et al. 2015). In particular, this is a problem for sequential data such as natural language data, which naturally possess a hierarchical generation process with complex intra-sequence dependencies. For instance, natural language dialogue has at least two levels of structure; within an utterance the structure is dominated by local statistics of the language (e.g. word co-occurrences), while across utterances there is a distinct source of variance characterized by aspects such as conversation topic and speaker goals. If a model only injects variability at the word level, it will have to decide on the conversation topic and speaker goals incrementally as it generates the words inside each utterance. This may lead to incoherent topics and inconsistent user goals (Pietquin and Hastie 2013).

We attack this problem in the natural language generation setting, specifically for (unstructured) dialogue response generation. Given a dialogue context in natural language, the model is tasked with generating an appropriate response word by word. This task has been investigated recently by many researchers using the sequence-to-sequence framework (Ritter, Cherry, and Dolan 2011; Lowe et al. 2015; Sordoni et al. 2015b; Li et al. 2016; Serban et al. 2016). Such models are not specifically designed for the goal-oriented setting, in which dialogue systems were originally developed (Gorin, Riccardi, and Wright 1997; Young 2000; Singh et al. 2002; Young et al. 2013). Nevertheless, major software companies are now developing non-goal-oriented models, which daily interact with millions of people. Two examples are Microsoft's Xiaolice (Markoff and Mozur 2015) and Google's Smart Reply system (Kannan et al. 2016), which at its core uses a sequence-to-sequence model. Currently, these models do not incorporate a hierarchical generation structure. Consequently, they cannot represent higher level variability and often fail to generate meaningful, diverse on-topic responses (Li et al. 2016).

Motivated by these shortcomings, we develop a hierarchical latent variable RNN architecture to explicitly model generative processes with multiple levels of variability. The model is a hierarchical sequence-to-sequence model with a continuous high-dimensional latent variable attached to each dialogue utterance, trained by maximizing a variational lower bound on the log-likelihood. In order to generate a response, the model first generates a sample of the continuous latent variable – representing the high-level semantic content of the response – and then it generates the response word by word conditioned on the latent variable. We apply the model to generate responses for Twitter conversations (Ritter, Cherry, and Dolan 2011; Sordoni et al. 2015b; Li et al. 2016). We evaluate the model and compare it to competing models through manual inspection and quantitatively

*A. S. was at Univerity of Montreal when this work was carried out.

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

using a human evaluation study on Amazon Mechanical Turk. The results demonstrate that the model substantially improves upon earlier models, and further highlight how the latent variables facilitate the generation of long utterances, with higher information content, and maintain dialogue context.

Technical Background

Recurrent Neural Network Language Model

A recurrent neural network (RNN), with parameters θ, models a variable-length sequence of tokens $(w_1, \ldots, w_M)$ by decomposing the probability distribution over outputs:

$$P_\theta(w_1, \ldots, w_M) = \prod_{m=2}^{M} P_\theta(w_m \mid w_1, \ldots, w_{m-1}) P_\theta(w_1). \tag{1}$$

The model processes each observation recursively. At each time step, the model observes an element and updates its internal hidden state, $h_m = f_\theta(h_{m-1}, w_m)$, where f is a parametrized non-linear function, called the activation or gating function, such as the hyperbolic tangent, the LSTM gating unit (Hochreiter and Schmidhuber 1997) or the GRU gating unit (Cho and others 2014). The hidden state summarizes the past sequence and parametrizes the output distribution of the model: $P_\theta(w_{m+1} \mid w_1, \ldots, w_m) = P_\theta(w_{m+1} \mid h_m)$. We assume the outputs lie within a discrete vocabulary V. Under this assumption the RNN Language Model (RNNLM) (Mikolov and others 2010)—one of the simplest generative RNN models for discrete sequences—parametrizes the output distribution using the softmax function applied to an affine transformation of the hidden state h_m. The model parameters are learned by maximizing the training log-likelihood using gradient descent.

Hierarchical Recurrent Encoder-Decoder (HRED)

The hierarchical recurrent encoder-decoder model (HRED) (Sordoni et al. 2015a; Serban et al. 2016) is an extension of the RNNLM. It generalizes the encoder-decoder architecture (Cho and others 2014) to the dialogue setting. HRED models each output sequence with a two-level hierarchy: a sequence of sub-sequences, and sub-sequences of tokens. In particular, a dialogue is modelled as a sequence of utterances (sub-sequences), with each utterance being a sequence of words:

$$\begin{aligned} P_\theta(\mathbf{w}_1, \ldots, \mathbf{w}_N) &= \prod_{n=1}^{N} P_\theta(\mathbf{w}_n | \mathbf{w}_{<n}), \\ &= \prod_{n=1}^{N} \prod_{m=1}^{M_n} P_\theta(w_{n,m} | w_{n,<m}, \mathbf{w}_{<n}), \end{aligned} \tag{2}$$

where, $\mathbf{w}_n$ is the n'th utterance in a dialogue, $w_{n,m}$ is the m'th word in the n'th utterance, and M_n is the number of words in the n'th utterance. HRED consists of three RNN modules: an *encoder* RNN, a *context* RNN and a *decoder* RNN. Each utterance is deterministically encoded into a real-valued vector by the *encoder* RNN:

$$h^{\text{enc}}_{n,0} = \mathbf{0}, \;\; h^{\text{enc}}_{n,m} = f^{\text{enc}}_\theta(h^{\text{enc}}_{n,m-1}, w_{n,m}) \; \forall m = 1, \ldots, M_n,$$

where f^{enc}_θ is either a GRU or a bidirectional GRU function. The last hidden state of the *encoder* RNN is given as input to the *context* RNN, which updates its internal hidden state to reflect all the information up until that utterance:

$$h^{\text{con}}_0 = \mathbf{0}, \;\; h^{\text{con}}_n = f^{\text{con}}_\theta(h^{\text{con}}_{n-1}, h^{\text{enc}}_{n,M_n}),$$

where f^{con}_θ is a GRU function taking as input two vectors. This hidden state is given to the *decoder* RNN:

$$\begin{gathered} h^{\text{dec}}_{n,0} = \mathbf{0}, \; h^{\text{dec}}_{n,m} = f^{\text{dec}}_\theta(h^{\text{dec}}_{n,m-1}, w_{n,m}, h^{\text{con}}_{n-1}) \\ \forall m = 1, \ldots, M_n, \end{gathered}$$

where f^{dec}_θ is the LSTM gating function taking as input three vectors. The output distribution is given by transforming $h^{\text{dec}}_{n,m}$ through a one-layer neural network (MLP) f^{mlp}_θ followed by an affine transformation and the softmax function:

$$P_\theta(w_{n,m+1} | w_{n,\leq m}, \mathbf{w}_{<n}) = \frac{e^{O^{\text{T}}_{w_{n,m+1}} f^{\text{mlp}}_\theta(h^{\text{dec}}_{n,m})}}{\sum_{w'} e^{O^{\text{T}}_{w'} f^{\text{mlp}}_\theta(h^{\text{dec}}_{n,m})}}, \tag{3}$$

where $O \in \mathbb{R}^{|V| \times d}$ is the word embedding matrix for the output distribution with embedding dimensionality $d \in \mathbb{N}$.

The Restricted Shallow Generation Process It has been observed that RNNLM and HRED, and similar models based on RNN architectures, have critical problems generating meaningful and diverse dialogue responses (Serban et al. 2016; Li et al. 2016). We believe these problems are caused by the flat sequential generation process followed by RNNLM and HRED, where each word is sampled conditioned only on previous words. We call this a *shallow* generation process, because the only source of variation is modelled through the conditional output distribution. This process is problematic from a probabilistic perspective, because the model is forced to generate all high-level structure locally on a step-by-step basis (Boulanger-Lewandowski, Bengio, and Vincent 2012; Bayer and Osendorfer 2014; Chung et al. 2015; Denton et al. 2015). For example, for generating dialogue responses such a model has to decide the conversation topic in the middle of the generation process – when it is generating the first topic-related word – and, afterwards, for each future word the model will have to decide whether to change or to remain on the same topic. This makes it difficult for the model to generate long-term structure. The shallow generation process is also problematic from a computational learning perspective: the state h_m in the RNNLM—or correspondingly the state of the *decoder* RNN in HRED—has to summarize all the past information up to time step m in order to (a) generate a probable next token (short-term objective) and (b) occupy a position in embedding space which sustains an output trajectory, for generating probable future tokens (long-term objective). Due to the vanishing gradient effect, the short-term goals will dominate the output distribution (Bengio, Simard, and Frasconi 1994). In particular, for sequences with high variability, the models are likely to favour short-term predictions as opposed to long-term predictions, because it is easier to only learn h_m for predicting the next token compared to sustaining a long-term trajectory $h_m, h_{m+1}, h_{m+2}, \ldots$, which at every time step is perturbed by noisy inputs (e.g. words given as input).

Latent Variable Hierarchical Recurrent Encoder-Decoder (VHRED)

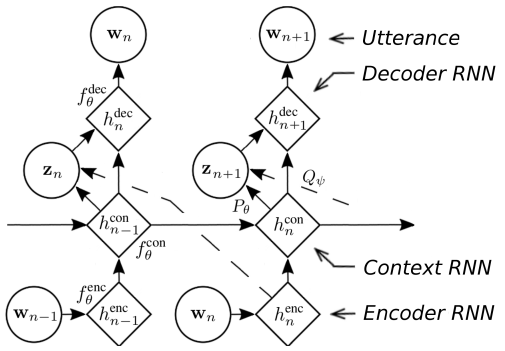

Figure 1: VHRED computational graph. Diamond boxes represent deterministic variables and rounded boxes represent stochastic variables. Full lines represent the generative model and dashed lines represent the approximate posterior model.

Motivated by the restricted shallow generation process, we propose the latent variable hierarchical recurrent encoder-decoder (VHRED) model. This model augments the HRED model with a stochastic latent variable at the utterance level, which is trained by maximizing a variational lower-bound on the log-likelihood. This allows it to model hierarchically-structured sequences in a two-step generation process—first sampling the latent variable, and then generating the output sequence—while maintaining long-term context.

VHRED contains a continuous high-dimensional stochastic latent variable $\mathbf{z}_n \in \mathbb{R}^{d_z}$ for each utterance $n = 1, \ldots, N$, which is conditioned on all the previous observed tokens. The model generates the n'th utterance tokens $\mathbf{w}_n$ through a two-level hierarchical generation process:

$$P_\theta(\mathbf{z}_n \mid \mathbf{w}_{<n}) = \mathcal{N}(\boldsymbol{\mu}_{\text{prior}}(\mathbf{w}_{<n}), \Sigma_{\text{prior}}(\mathbf{w}_{<n})),$$

$$P_\theta(\mathbf{w}_n \mid \mathbf{z}_n, \mathbf{w}_{<n}) = \prod_{m=1}^{M_n} P_\theta(w_{n,m} \mid \mathbf{z}_n, \mathbf{w}_{<n}, w_{n,<m}),$$

where $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is the multivariate normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^{d_z}$ and covariance matrix $\Sigma \in \mathbb{R}^{d_z \times d_z}$, which is constrained to be a diagonal matrix.

VHRED (Figure 1) contains the same three components as the HRED model. The *encoder* RNN deterministically encodes a single utterance into a fixed-size real-valued vector, which the *context* RNN takes as input in order to compute its hidden state h_n^{con} for the n'th utterance. The vector h_n^{con} is transformed through a two-layer feed-forward neural network with hyperbolic tangent gating function. A matrix multiplication is applied to the output of the feed-forward network, which defines the multivariate normal mean $\boldsymbol{\mu}_{\text{prior}}$. Similarly, for the diagonal covariance matrix Σ_{prior} a different matrix multiplication is applied to the net's output followed by softplus function, to ensure positiveness (Chung et al. 2015).

The model's latent variables are inferred by maximizing the variational lower-bound, which factorizes into independent terms for each sub-sequence (utterance):

$$\begin{aligned}
&\log P_\theta(\mathbf{w}_1, \ldots, \mathbf{w}_N) \\
&\geq \sum_{n=1}^{N} -\text{KL}\left[Q_\psi(\mathbf{z}_n \mid \mathbf{w}_1, \ldots, \mathbf{w}_n) || P_\theta(\mathbf{z}_n \mid \mathbf{w}_{<n})\right] \\
&+ \mathbb{E}_{Q_\psi(\mathbf{z}_n \mid \mathbf{w}_1, \ldots, \mathbf{w}_n)}\left[\log P_\theta(\mathbf{w}_n \mid \mathbf{z}_n, \mathbf{w}_{<n})\right], \qquad (4)
\end{aligned}$$

where $\text{KL}[Q||P]$ is the Kullback-Leibler (KL) divergence between distributions Q and P. The distribution Q_ψ is the approximate posterior distribution – also known as the *encoder model* or *recognition model* – which approximates the intractable true posterior distribution:

$$\begin{aligned}
&Q_\psi(\mathbf{z}_n \mid \mathbf{w}_1, \ldots, \mathbf{w}_N) \\
&= \mathcal{N}(\boldsymbol{\mu}_{\text{posterior}}(\mathbf{w}_1, \ldots, \mathbf{w}_n), \Sigma_{\text{posterior}}(\mathbf{w}_1, \ldots, \mathbf{w}_n)) \\
&\approx P_\psi(\mathbf{z}_n \mid \mathbf{w}_1, \ldots, \mathbf{w}_N), \qquad (5)
\end{aligned}$$

where $\boldsymbol{\mu}_{\text{posterior}}$ and $\Sigma_{\text{posterior}}$ respectively define the approximate posterior mean and posterior covariance matrix (assumed diagonal) as a function of the previous utterances $\mathbf{w}_1, \ldots, \mathbf{w}_{n-1}$ and the current utterance $\mathbf{w}_n$. The posterior mean $\boldsymbol{\mu}_{\text{posterior}}$ and covariance $\Sigma_{\text{posterior}}$ are determined similar to the prior. At the n'th utterance, a feed-forward network takes as input the concatenation of both h_n^{con} (summary of past utterances) and $h_{n+1,M_{n+1}}^{\text{enc}}$ (current utterance summary). The network's output is transformed through a matrix multiplication to give the mean, and by matrix multiplication and a softplus function to give the diagonal covariance matrix.

At generation time, the model conditions on the previous observed utterances and draws $\mathbf{z}_n$ from the prior $\mathcal{N}(\boldsymbol{\mu}_{\text{prior}}(\mathbf{w}_{<n}), \Sigma_{\text{prior}}(\mathbf{w}_{<n}))$. The sample and the output of the *context* RNN are given as input to the *decoder* RNN:

$$h_{n,0}^{\text{dec}} = \mathbf{0}, \; h_{n,m}^{\text{dec}} = f_\theta^{\text{dec}}(h_{n,m-1}^{\text{dec}}, w_{n,m}, h_{n-1}^{\text{con}}, \mathbf{z}_n) \quad \forall m = 1, \ldots, M_n,$$

and the output tokens are sampled according to eq. (3). When training the model, for each utterance in a training example $n = 1, \ldots, N$, a sample $\mathbf{z}_n$ is drawn from the approximate posterior $\mathcal{N}(\boldsymbol{\mu}_{\text{posterior}}(\mathbf{w}_1, \ldots, \mathbf{w}_n), \Sigma_{\text{posterior}}(\mathbf{w}_1, \ldots, \mathbf{w}_n))$. This sample is used to estimate the gradient w.r.t. the variational lower-bound given by eq. (4).

As will be shown in the next section, VHRED alleviates the problems arising from the insufficient shallow generation process followed by the RNNLM and HRED models. The variation of the output sequence is now modelled in two ways: at the utterance-level (sequence-level) with the conditional prior distribution over $\mathbf{z}$, and at the word-level (sub-sequence-level) with the conditional distribution over word tokens. The effect of the variable $\mathbf{z}$ corresponds to higher-level decisions about what to generate, like the conversation topic, speaker goals or sentiment of the utterance. By representing high-level information about the sequence, $\mathbf{z}$ helps model long-term output trajectories. This allows the *decoder* RNN hidden state to focus only on summarizing the current utterance.

Alternative Architectures In the course of developing the VHRED architecture we considered different variants. We experimented with a model where the *context* RNN hidden state

h_n^{con} was not given as input to the *decoder* RNN. However, this architecture performed worse because all the context information had to be passed through the latent variable $\mathbf{z}_n$, which effectively made $\mathbf{z}_n$ an information bottleneck. We also experimented with a variant where the mean of the latent variable $\mathbf{z}_n$ would depend on the mean of the previous latent variable $\mathbf{z}_{n-1}$. However, this destabilized the training process. Lastly, we experimented with a variant, where the posterior distribution for $\mathbf{z}_n$ would also be conditioned on the future *context* RNN states h_{n+1}^{con}. This additional information did not improve performance w.r.t. the variational lower bound.

Experimental Evaluation

We apply VHRED to dialogue response generation. Given a dialogue context, the model must generate an appropriate response. This task has been studied extensively in the recent literature (Ritter, Cherry, and Dolan 2011; Lowe et al. 2015; Sordoni et al. 2015b; Li et al. 2016; Serban et al. 2016).

We experiment on the **Twitter Dialogue Corpus** (Ritter, Cherry, and Dolan 2011). The task is to generate utterances to append to existing Twitter conversations. The dataset is extracted using a procedure similar to Ritter et al. (2011), and is split into training, validation and test sets, containing respectively 749,060, 93,633 and 10,000 dialogues each.[1] Each dialogue contains on average 6.27 utterances and 94.16 words. The dialogues are substantially longer than recent large-scale language modelling corpora, such as the 1 Billion Word Language Model Benchmark (Chelba et al. 2014), which focus on modelling single sentences.

Training and Evaluation Procedures

We implement all models using Theano (Theano Development Team 2016). We optimize all models using Adam (Kingma and Ba 2015). We early stop and select hyperparameters using the variational lower-bound or log-likelihood on the validation set. At test time, we use beam search with 5 beams for outputting responses with the RNN decoders (Graves 2012). For the VHRED models, we sample the latent variable $\mathbf{z}_n$, and condition on it when executing beam search with the RNN decoder. We use word embedding dimensionality of size 400. All models were trained with a learning rate of 0.0001 or 0.0002 and with mini-batches containing 40 or 80 training examples. We use truncated back-propagation and gradient clipping.

Baselines We compare to an LSTM model with 2000 hidden units. The architecture was chosen w.r.t. validation set log-likelihood. We also compare to the HRED model. The HRED model *encoder* RNN is a bidirectional GRU RNN encoder, where the forward and backward RNNs each have 1000 hidden units. The *context* RNN and *decoder* RNN have each 1000 hidden units. This architecture performed best in preliminary experiments w.r.t. validation set log-likelihood. Both the LSTM and HRED models have previously been proposed for dialogue response generation (Serban et al. 2016; Vinyals and Le 2015). For reference with earlier work not based on neural networks, we also compare to the TF-IDF retrieval model (Lowe et al. 2015).

[1]The Twitter tweet IDs will be made available upon publication.

VHRED The *encoder* and *context* RNNs for VHRED are parametrized in the same way as the corresponding HRED model. The only difference is in the parametrization of the *decoder* RNN, which takes as input the *context* RNN output vector concatenated with the generated stochastic latent variable. Furthermore, we initialize the feed-forward networks of the prior and posterior distributions with values drawn from a zero-mean normal distribution with variance 0.01 and with biases equal to zero. We also multiply the diagonal covariance matrices of the prior and posterior distributions with 0.1 to make training more stable, because a high variance makes the gradients w.r.t. the reconstruction cost unreliable, which is fatal at the beginning of training.

Further, VHRED's *encoder* and *context* RNNs are initialized to the parameters of the converged HRED model. We use the two heuristics proposed by Bowman et al. (2016): we drop words in the decoder with a fixed drop rate of 25% and multiply the KL terms in eq. (4) by a scalar, which starts at zero and linearly increases to 1 over the first 60,000 training batches. Applying these heuristics helped substantially to stabilize the training process and improve the learned representations of the stochastic latent variables. We also experimented with the batch normalization training procedure for the feed-forward neural networks. but found that this made training very unstable without any substantial gains in performance w.r.t. the variational bound.

Human Evaluation Evaluation of dialogue system responses is a difficult and open problem (Galley and others 2015; Pietquin and Hastie 2013). Inspired by metrics used for evaluating machine translation and information retrieval systems, researchers have begun adopting word-overlap metrics such as BLEU. However, Liu et al. (2016) show that such metrics have little correlation with human evaluations of response quality. Similarly, metrics such as word perplexity have also been criticized as inappropriate for evaluation (Pietquin and Hastie 2013). We therefore conduct a human evaluation to compare the responses from different models.

We carry out the human study on Amazon Mechanical Turk (AMT). Our setup follows that of Sordoni et al. (2015b). We show human evaluators a dialogue context along with two potential responses: one response generated from each model (generated conditioned on dialogue context). We ask evaluators to choose the response most appropriate to the dialogue context. If the evaluators are indifferent to either of the two responses, or if they cannot understand the dialogue context, they can choose neither response. For each pair of models we conduct two experiments: one where the example contexts contain at least 80 unique tokens (*long context*), and one where they contain at least 20 (not necessarily unique) tokens (*short context*). This helps compare how well each model can integrate the dialogue context into its response, since it has previously been argued that for long contexts hierarchical RNNs models fare better (Sordoni et al. 2015a).

The results (Table 1) show that VHRED is clearly preferred in the majority of the experiments. In particular, VHRED is strongly preferred over the HRED and TF-IDF baseline models for both short and long context settings. VHRED is also strongly preferred over the LSTM baseline model for long contexts, although the LSTM model is preferred over

Table 2: Twitter examples for the neural network models. The → token indicates a change of turn.

Context	Response
jamaica blood!! <url> → lol. yes [name]!. [name]! → what's poppin in toronto homie	**VHRED:** nothing much. just chillin. what's good with you? **LSTM:** something!!!!!!! **HRED:** something!!!!!!
reaches for [name]! → - approaches him warily - hey there buddy. - swings him into the air - how you doin? → tilts his head and your wariness and reaches for your face [name].	**VHRED:** - looks up at you and smiles - you're so cute. - kisses your cheek - **LSTM:** - smiles - i'm fine thank you. how are you? **HRED:** - grins, looking at you as i look at you -
my math teacher over here talkn bout nickels nd i'm tryna make big money i need to be in calculus → lol	**VHRED:** i need to go to the gym and get my hair done lol **LSTM:** lol **HRED:** how u been
i think i want an ipad2. but i already have an ibook. want vs. need. → if you have an ibook. then you deserve an ipad!	**VHRED:** that's what i'm thinking. i'm going to buy a new one. **LSTM:** i want one!!!! **HRED:** i don't know what to do with it. i don't know what to do.
ah echt? dan anrtwoord ik volgende keer via twitter als je me smst. → haha doe maar p. goed bezig hoor myrthe d → goed bezig? die snap ik even niet p	**VHRED:** haha, ik heb er geen zin in de something, maar ik heb er geen something meer something **LSTM:** haha something something something **HRED:** something something something something something something something something

Table 1: Wins, losses and ties (in %) of VHRED against baselines based on the human study (mean preferences ± 90% confidence intervals, where ∗ indicates significant differences at 90% confidence)

Opponent	Wins	Losses	Ties
Short Contexts			
VHRED vs LSTM	32.3 ±2.4	**42.5 ±2.6**∗	25.2 ±2.3
VHRED vs HRED	**42.0 ±2.8**∗	31.9 ±2.6	26.2 ±2.5
VHRED vs TF-IDF	**51.6 ±3.3**∗	17.9 ±2.5	30.4 ±3.0
Long Contexts			
VHRED vs LSTM	**41.9 ±2.2**∗	36.8 ±2.2	21.3 ±1.9
VHRED vs HRED	**41.5 ±2.8**∗	29.4 ±2.6	29.1 ±2.6
VHRED vs TF-IDF	**47.9 ±3.4**∗	11.7 ±2.2	40.3 ±3.4

Table 3: Response evaluation using topic similarity metrics (Average, Greedy and Extrema) and entropy metrics (entropy per word H_w, and entropy per utterance H_U).

Model	Average	Greedy	Extrema	H_w	H_U
LSTM	0.512	0.389	0.366	6.75	75.61
HRED	0.501	0.378	0.355	6.73	78.35
VHRED	**0.533**	**0.396**	**0.38**	**6.88**	**84.56**

VHRED for short contexts. In conclusion, VHRED performs substantially better overall than competing models.

For short contexts, the LSTM model is often preferred over VHRED because the LSTM model tends to generate very *generic* responses (see Table 3). This behaviour was also reported in previous work (Serban et al. 2016; Li et al. 2016). Such *generic* or *safe* responses are reasonable for a wide range of contexts. Thus, human evaluators are more likely to rate them as appropriate compared to semantically richer responses (e.g. responses related to a specific topic) when the context is short. However, a model that only outputs generic responses is generally undesirable for dialogue as it

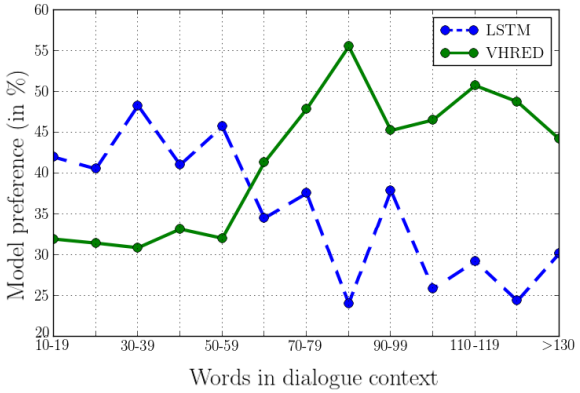

Figure 2: Human evaluator preferences for VHRED vs LSTM by context length excluding ties. For short contexts humans prefer the generic responses generated by LSTM, while for long contexts humans prefer the semantically richer responses generated by VHRED.

leads to less engaging and meaningless conversations. On the other hand, VHRED is explicitly designed for incorporating long contexts and for outputting a diverse set of responses by sampling of the latent variable. Thus, VHRED generates longer sentences with more semantic content than the LSTM model (see Table 3). This can be *riskier* as longer utterances are more likely to contain small mistakes, which can lead to lower human preference for a single utterance. However, response diversity is crucial for maintaining interesting conversations (Shaikh et al. 2010). This conclusion is supported by examination of the human preferences w.r.t. context length (see Figure 2), which shows human preferences for VHRED increase as the dialogue contexts become longer.

Metric-based Evaluation To evaluate how semantically relevant and on-topic the responses are, we further report results for three word embedding-based topic similarity metrics proposed by Liu et al. (2016): *Embedding Average* (Aver-

age), *Embedding Extrema* (Extrema) and *Embedding Greedy* (Greedy) (Mitchell and Lapata 2008; Forgues et al. 2014; Rus and Lintean 2012).[2] To analyze the information content of the responses, we also report average entropy (in bits) – w.r.t. the maximum likelihood unigram model over the generated responses – per word and per response.[3]

The results are given in Table 3. According to the topic similarity metrics, VHRED responses are substantially more on-topic compared to LSTM and HRED. According to the entropy metrics, VHRED responses also contain substantially more information content. In comparison to the generic responses of the baseline models (Serban et al. 2016; Li et al. 2016), this suggests the hierarchical generation process facilitates the generation of more on-topic responses, as well as semantically diverse and meaningful responses. This indicates the VHRED hidden states traverse a larger area of the semantic space compared to the HRED and LSTM.

Qualitative Evaluation The conclusions above are also supported by a qualitative assessment of the generated responses. In the examples shown in Table 2. we see that VHRED has learned to better model smilies and slang (first example in Table 2). Furthermore, VHRED appears to be better at generating *stories* and *imaginative actions* compared to competing models (second example in Table 2). The third example in Table 2 is a case where VHRED generated response is more interesting, yet may be less preferred by humans as it is slightly incompatible with the context, compared to the generic LSTM response – although topic switches do occur frequently. Finally, VHRED is able to continue conversations in different languages (fifth example in Table 2). This came as a surprise to us, because we had preprocessed the dataset by filtering out non-English tweets. VHRED, however, learned to distinguish between English, Spanish and Dutch conversations from the remaining non-English tweets in the preprocessed dataset. Such aspects are not measured by the human study, but are evident in our qualitative inspection.

In support of these findings, we also carried out an analysis of the latent representations learned by VHRED. Our analysis showed VHRED has learned to separate different types of responses, such as *how are you* and *thank you* responses, and responses with Spanish or Dutch language.

Related Work

Previous research on dialogue models has investigated latent variable models. Zhai and Williams (2014) propose three models combining hidden Markov models and topic models. Unlike VHRED, these models were developed solely for learning representations – not for generating responses. Learning latent representations for dialogue was also pursued by Bangalore et al. (2008), by Crook et al. (2009) and others.

The use of a stochastic latent variable learned by maximizing a variational lower bound is inspired by the variational autoencoder (VAE) (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014). Such models have been used predominantly for generating images in the continuous domain (Gregor et al. 2015). However, there has also been recent work applying these architectures for generating sequences, such as the Variational Recurrent Neural Networks (VRNN) (Chung et al. 2015), which was applied for speech and handwriting synthesis, and Stochastic Recurrent Networks (STORN) (Bayer and Osendorfer 2014), which was applied for music generation and motion capture modelling. Unlike VHRED, these models sample a separate latent variable at each time step of the decoder; they do not exploit hierarchical structure for modelling higher-level variability.

Most similar to our work is the Variational Recurrent Autoencoder (VRAE) (Fabius and van Amersfoort 2015) and the Variational Autoencoder Language Model(VAELM) (Bowman et al. 2016), which apply encoder-decoder architectures to generative music modelling and language modelling respectively. Unlike VRAE and VAELM, the VHRED latent variable is conditioned on all previous utterances. This makes the latent variables co-dependent through the observed tokens, but also enables VHRED to generate multiple utterances on the same topic. Further, VHRED uses a hierarchical architecture similar to the HRED model, which enables it to model long-term context. It also has a direct deterministic connection between the *context* and *decoder* RNN, which allows the model to transfer deterministic pieces of information between its components. Finally, VHRED achieves improved results beyond the autoencoder framework, where the objective is conditional generation.

Conclusion

Current sequence-to-sequence models for dialogue response generation follow a shallow generation process, which limits their ability to model high-level variability. Consequently, these models fail to generate meaningful and diverse on-topic responses. To overcome these problems, we have introduced a hierarchical latent variable neural network architecture, called VHRED. VHRED uses a hierarchical generation process in order to exploit the with-in sequence structure in utterances and is trained using a variational lower bound on the log-likelihood. We have applied VHRED to the task of dialogue response generation, where it yields a substantial improvement over competing models in several ways, including quality of responses as measured in a human evaluation study. The empirical results highlight the advantages of the hierarchical generation process for generating meaningful and diverse on-topic responses.

The proposed model can easily be extended to several other sequential generation tasks that exhibit a hierarchical structure, such as document-level machine translation, web query prediction, music composition, multi-sentence document summarization and image caption generation.

Acknowledgments The authors thank Michael Noseworthy and Sungjin Ahn for constructive feedback. The authors acknowledge NSERC, Canada Research Chairs and CIFAR for funding. Ryan Lowe and Joelle Pineau were funded by the Samsung Advanced Institute of Technology (SAIT). This research was enabled in part by support provided by Calcul Qubec (`www.calculquebec.ca`) and Compute Canada

[2] We use the embeddings trained on Google News Corpus: `https://code.google.com/archive/p/word2vec/`.

[3] The unigram model is trained on the training set, and the entropy is computed on the preprocessed tokenized dataset.

(www.computecanada.ca).

References

Bangalore, S.; Di Fabbrizio, G.; and Stent, A. 2008. Learning the structure of task-driven human–human dialogs. *IEEE Transactions on Audio, Speech, and Language Processing* 16(7):1249–1259.

Bayer, J., and Osendorfer, C. 2014. Learning stochastic recurrent networks. In *NIPS, Workshop on Advances in Variational Inference*.

Bengio, Y.; Simard, P.; and Frasconi, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5(2):157–166.

Boulanger-Lewandowski, N.; Bengio, Y.; and Vincent, P. 2012. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *ICML*.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Jozefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. *CoNLL*.

Chelba, C.; Mikolov, T.; Schuster, M.; Ge, Q.; Brants, T.; Koehn, P.; and Robinson, T. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH*.

Cho, K., et al. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*.

Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A.; and Bengio, Y. 2015. A recurrent latent variable model for sequential data. In *NIPS*.

Crook, N.; Granell, R.; and Pulman, S. 2009. Unsupervised classification of dialogue acts using a dirichlet process mixture model. In *SIGDIAL*.

Denton, E. L.; Chintala, S.; Szlam, A.; and Fergus, R. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*.

Fabius, O., and van Amersfoort, J. R. 2015. Variational recurrent auto-encoders. *ICLR, Workshop Papers*.

Forgues, G.; Pineau, J.; Larchevêque, J.-M.; and Tremblay, R. 2014. Bootstrapping dialog systems with word embeddings. *NIPS, Modern Machine Learning and Natural Language Processing Workshop*.

Galley, M., et al. 2015. deltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. In *ACL*.

Goodfellow, I.; Courville, A.; and Bengio, Y. 2015. *Deep Learning*. MIT Press.

Gorin, A. L.; Riccardi, G.; and Wright, J. H. 1997. How may i help you? *Speech communication* 23(1):113–127.

Graves, A. 2012. Sequence transduction with recurrent neural networks. In *ICML, Representation Learning Workshop*.

Gregor, K.; Danihelka, I.; Graves, A.; and Wierstra, D. 2015. DRAW: A recurrent neural network for image generation. In *ICLR*.

Hinton, G., et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE* 29(6):82–97.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8).

Kannan, A.; Kurach, K.; Ravi, S.; Kaufmann, T.; Tomkins, A.; Miklos, B.; Corrado, G.; Lukács, L.; Ganea, M.; et al. 2016. Smart reply: Automated response suggestion for email. In *ACM SIGKDD*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *ICLR*.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL*.

Liu, C.-W.; Lowe, R.; Serban, I. V.; Noseworthy, M.; Charlin, L.; and Pineau, J. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*.

Lowe, R.; Pow, N.; Serban, I.; and Pineau, J. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *SIGDIAL*.

Markoff, J., and Mozur, P. 2015. For sympathetic ear, more chinese turn to smartphone program. *NY Times*.

Mikolov, T., et al. 2010. Recurrent neural network based language model. In *INTERSPEECH*.

Mitchell, J., and Lapata, M. 2008. Vector-based models of semantic composition. In *ACL*, 236–244.

Pietquin, O., and Hastie, H. 2013. A survey on metrics for the evaluation of user simulations. *The knowledge engineering review* 28(01):59–73.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*.

Ritter, A.; Cherry, C.; and Dolan, W. B. 2011. Data-driven response generation in social media. In *EMNLP*.

Rus, V., and Lintean, M. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *ACL, Building Educational Applications Workshop*.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*.

Shaikh, S.; Strzalkowski, T.; Taylor, S.; and Webb, N. 2010. VCA: an experiment with a multiparty virtual chat agent. In *ACL, Workshop on Companionable Dialogue Systems*.

Singh, S.; Litman, D.; Kearns, M.; and Walker, M. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *JAIR* 16:105–133.

Sordoni, A.; Bengio, Y.; Vahabi, H.; Lioma, C.; Simonsen, J. G.; and Nie, J.-Y. 2015a. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *CIKM*.

Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.-Y.; Gao, J.; and Dolan, B. 2015b. A neural network approach to context-sensitive generation of conversational responses. In *NAACL-HLT*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688.

Vinyals, O., and Le, Q. 2015. A neural conversational model. *ICML, Deep Learning Workshop*.

Young, S.; Gasic, M.; Thomson, B.; and Williams, J. D. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE* 101(5):1160–1179.

Young, S. 2000. Probabilistic methods in spoken–dialogue systems. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 358(1769).

Zhai, K., and Williams, J. D. 2014. Discovering latent structure in task-oriented dialogues. In *ACL*.