A Multi-Level Latent Variable Encoder-Decoder Framework for Dialogue Generation

TONY LIVINS

Abstract

Sequential datasets often display multi-layered structure, with meaningful dependencies appearing not only within segments but also across larger subsequences a characteristic clearly present in multi-turn dialogue. To model this style of generative behaviour, we introduce a neural generative framework that incorporates stochastic latent variables operating across flexible temporal ranges. We apply the model to dialogue response generation and compare its performance with several recently proposed neural architectures. Evaluation is conducted using both automated metrics and structured human assessments. Experimental results indicate that our approach surpasses current baselines and that the latent representations enable the production of longer, more contextually consistent responses.

1 Introduction

Deep recurrent neural networks (RNNs) have shown remarkable progress on challenging tasks requiring the generation of organised sequential outputs [9]. Their effectiveness has been demonstrated across a wide range of areas, including language modelling [10, 18], machine translation [28, 5], conversational modelling [27, 24], and speech recognition [9].

Although these achievements are substantial, traditional RNNs typically incorporate a simple source of variability: randomness is introduced only when sampling output tokens. Prior research suggests that placing all stochastic behaviour at the output layer is often inadequate [2, 6, 1]. This limitation becomes especially relevant for sequential domains such as natural language and speech, where the underlying generative mechanisms are hierarchical and involve complex dependencies. Dialogue, for example, contains at least two structured layers. Individual utterances are governed by local linguistic statistics, while transitions between utterances involve higher-level uncertainties influenced by factors such as topic, speaker intention, and conversational style.

In this work, we present a hierarchical stochastic latent-variable neural architecture designed to capture generative processes that exhibit multiple sources of variation. We examine the model on the task of dialogue response generation and benchmark it against recent neural systems. Our evaluation includes qualitative inspection, automatic scoring methods, and human assessments conducted through Amazon Mechanical Turk.

2 Technical Background

2.1 Recurrent Neural Network Language Model

A recurrent neural network (RNN) with parameters represents a variable-length token sequence
$(w_1, \ldots, w_M)$by factorizing the joint probability of the sequence as:

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

At each time step, the network processes the current token and updates its hidden state according to
$h_m = f(h_{m-1}, w_m)$, where $f$is a nonlinear transformation such as a tanh unit, an LSTM [12], or a GRU [5].1
The hidden state functions as a compact summary of all preceding information and parametrizes the distribution over the next token:

$$P_\theta(w_{m+1} \mid w_1, \ldots, w_m) = P_\theta(w_{m+1} \mid h_m)$$

Given that outputs are drawn from a discrete vocabulary V, the standard RNN Language Model (RNNLM) [18] defines the next-token distribution using a softmax transformation applied to an affine projection of the hidden state h_m. Parameters are trained by maximizing the log-likelihood of the training data through gradient descent.

2.2 Hierarchical Recurrent Encoder-Decoder

The hierarchical recurrent encoder–decoder model (HRED) [26, 24] expands the RNNLM by adapting the encoder–decoder framework [5] to multi-turn dialogue. The HRED assumes that an output sequence follows a two-level hierarchical structure: sequences made up of sub-sequences, and sub-sequences made up of individual tokens. For example, a dialogue can be viewed as a sequence of utterances (sub-sequences), each composed of a series of words. Likewise, a document can be interpreted as a sequence of sentences, with each sentence represented as a word sequence.

The HRED architecture is composed of three RNN components: an encoder RNN, a context RNN, and a decoder RNN. Each sub-sequence is transformed into a continuous vector representation by the encoder RNN. This representation is passed to the context RNN, which updates its hidden state to summarise all previously processed sub-sequences. The context RNN then outputs a continuous vector used by the decoder RNN to generate the next sub-sequence of tokens. Additional explanation can be found in [26, 24].

2.3 A Deficient Generation Process

Recent studies have pointed out that models such as the RNNLM, HRED, and similar RNN-based architectures struggle to produce coherent and meaningful dialogue utterances [24, 15]. We argue that the fundamental limitation stems from how these models parametrise their output distributions. Because variability is introduced only at the token-level output distribution, the generative process is overly constrained.

This limitation manifests in two ways. From a probabilistic standpoint, injecting stochasticity only at the lowest level forces the model to prioritise short-range patterns rather than broader, long-term dependencies. Low-level noise is tightly bound to the most recent context but only loosely influenced by older or future structure. In a more abstract sense, if one imagines variability introduced through i.i.d. noise added to deterministic components, noise applied at higher representational levels spanning multiple time steps would naturally capture longer-range dependencies.

From a learning perspective, the RNNLM's hidden state h_m(or the decoder state in HRED) is required to simultaneously satisfy two demanding objectives:
(a) generate a high-probability next token (short-term requirement), and
(b) maintain a representation that supports a plausible long-term trajectory of outputs.

This dual burden makes learning difficult because the model must encode both immediate predictive cues and long-range structural information within the same hidden state.

Predicting future tokens (the long-range objective) becomes disproportionately harder. Because of the vanishing-gradient phenomenon, the optimisation process naturally gives more weight to short-term objectives. As a result, the model tends to favour parameters that excel at predicting only the immediate next token. This tendency is especially strong for high-entropy sequences, where the easiest solution is to optimise the hidden state $h_m$strictly for next-token prediction rather than for sustaining a coherent long-term trajectory. Each time step is influenced by a noisy observation, making it much more difficult for the model to preserve stable long-range structure.

3 Latent Variable Hierarchical Recurrent Encoder-Decoder (VHRED)

Building on the issues noted above, we introduce the Variational Hierarchical Recurrent Encoder–Decoder (VHRED) model. VHRED extends the HRED architecture by incorporating a latent variable at each decoder step and training the system using a variational lower bound on the log-likelihood. This design enables the model to represent hierarchical sequences through a two-stage generation process: (1) sample a latent variable, and (2) produce the corresponding output sub-sequence, while preserving long-term contextual information.

Let $w_1, \dots, w_N$ denote a sequence of N sub-sequences, where the n-th sub-sequence is $w_n = (w_{n,1}, \dots, w_{n,M_n})$ and each token $w_{n,m} \in V$. The VHRED introduces a stochastic latent variable $z_n \in \mathbb{R}^{d_z}$ for each sub-sequence $n = 1, \dots, N$, conditioned on all previously observed tokens. Once z_n is sampled, the corresponding sub-sequence w_n is generated:

$$P_\theta(z_n \mid w_1, \dots, w_{n-1}) = \mathcal{N}(\mu_{\text{prior}}(w_1, \dots, w_{n-1}), \Sigma_{\text{prior}}(w_1, \dots, w_{n-1})) \quad (2)$$

$$P_\theta(w_n \mid z_n, w_1, \dots, w_{n-1}) = \prod_{m=1}^{M_n} P_\theta(w_{n,m} \mid z_n, w_1, \dots, w_{n-1}, w_{n,1}, \dots, w_{n,m-1}) \quad (3)$$

Here, $\mathcal{N}(\mu, \Sigma)$ is a multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^{d_z}$ and diagonal covariance $\Sigma \in \mathbb{R}^{d_z \times d_z}$.

The VHRED architecture (Figure 1) retains the same three modules as the HRED: an encoder RNN, a context RNN, and a decoder RNN. The encoder transforms each sub-sequence into a fixed-dimensional vector. The context RNN processes these vectors sequentially, maintaining a deterministic summary of all previous sub-sequences. Its hidden state is passed through a two-layer feed-forward network with a tanh activation. A linear projection of this network's output defines the prior mean μ_{prior}. A separate linear projection followed by a softplus activation produces the diagonal covariance Σ_{prior}, ensuring positivity [6].

To infer the latent variables, the model maximizes the variational lower bound, which decomposes across sub-sequences:

$$\log P_\theta(w_1, \dots, w_N) \geq \sum_{n=1}^{N} [-\text{KL}(Q(z_n \mid w_1, \dots, w_n) \parallel P_\theta(z_n \mid w_1, \dots, w_{n-1})) + \mathbb{E}_Q[\log P_\theta(w_n \mid z_n, w_1,$$

where $\text{KL}[Q \parallel P]$ denotes the Kullback–Leibler divergence.

The approximate posterior distribution (also called the recognition or encoder model) is defined as:

$$Q(z_n \mid w_1, \ldots, w_N) = Q(z_n \mid w_1, \ldots, w_n) = \mathcal{N}(\mu_{\text{post}}(w_1, \ldots, w_n), \Sigma_{\text{post}}(w_1, \ldots, w_n)) \propto P(z_n \mid w_1, \ldots, w_N)$$

Here, $\mu_{\text{post}}$and $\Sigma_{\text{post}}$are the approximate posterior mean and diagonal covariance, computed in the same way as the prior via linear transformations of the feed-forward network output, with softplus used for the covariance.

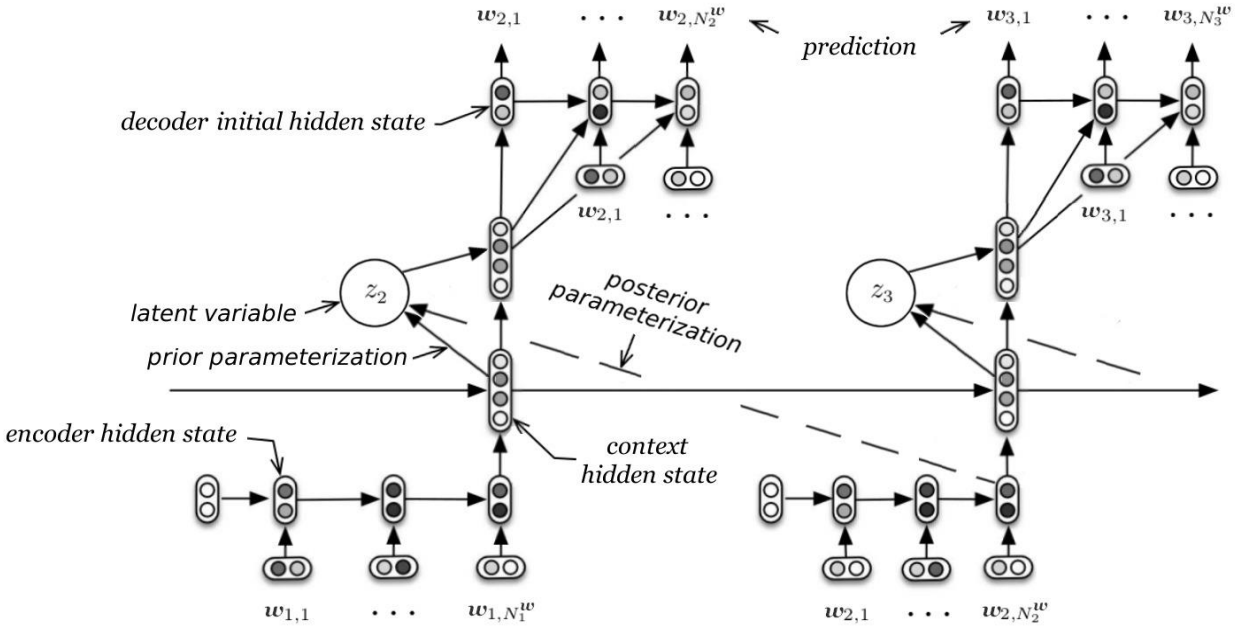

Figure 1Computational graph for the VHRED model. The rounded boxes indicate deterministic, real-valued vectors. The variables

At test time, conditioned on the previously observed sub-sequences $(w_1, \ldots, w_{n-1})$, a latent sample $z_n$is drawn from the prior distribution

$$\mathcal{N}(\mu_{\text{prior}}(w_1, \ldots, w_{n-1}), \Sigma_{\text{prior}}(w_1, \ldots, w_{n-1}))$$

for each sub-sequence. This sampled vector is concatenated with the output produced by the context RNN and then provided as input to the decoder RNN, following the same process as the HRED architecture. The decoder then generates the sub-sequence one token at a time.

During training, for each $n = 1, \ldots, N$, a latent sample $z_n$is drawn from the approximate posterior
$\mathcal{N}(\mu_{\text{post}}(w_1, \ldots, w_n), \Sigma_{\text{post}}(w_1, \ldots, w_n))$.

This sample is used to compute the gradient of the variational lower bound described in Equation (4). The approximate posterior has its own one-layer feed-forward neural network, which receives as input the context RNN state at the current step along with the encoder RNN output for the subsequent sub-sequence.

The VHRED model reduces several of the generation issues found in the RNNLM and HRED. Instead of relying on a single source of variability, the VHRED incorporates variation at two levels: the sequence level through the conditional prior over z, and the sub-sequence level through the conditional token distribution over $w_1, \ldots, w_M$. The latent variable $z$supports long-range output structure by encoding high-level properties of the sequence. This allows the hidden state $h_m$to concentrate on summarizing information up to token M. In simple terms, the randomness introduced by $z$captures higher-level factors such as the topic or emotional tone of the utterance.

4 Experimental Evaluation

We evaluate the model on the task of conditional natural language response generation in dialogue. This task is useful in settings such as customer support, technical help systems, language tutoring environments, and entertainment applications [29]. It also requires the ability to generate sequences with complex structure while maintaining awareness of long-term context [17, 27].

We consider two separate tasks. In both cases, the model receives a dialogue context consisting of one or more utterances, and the objective is to generate the next appropriate response. Our initial experiments use the Twitter Dialogue Corpus [22]. The goal is to produce utterances that can be appended to ongoing Twitter conversations. The dataset is constructed using a method similar to Ritter et al. [22], and is divided into training, validation, and test sets with 749,060, 93,633, and 10,000 dialogues respectively. Each dialogue contains an average of 6.27 utterances and 94.16 tokens.2

These conversations are considerably longer than those in many large-scale language modelling benchmarks, such as the One Billion Word Benchmark [4], which focus on predicting single sentences.

2 Due to Twitter's terms of service, the actual tweet text cannot be redistributed. Only tweet IDs can be released publicly.

We also run experiments on the Ubuntu Dialogue Corpus [17], which contains approximately 500,000 dialogues collected from the #Ubuntu chat channel on Internet Relay Chat. Users typically join the channel with a technical question related to Ubuntu, and other users respond with advice. Additional information is provided in Appendix

6.1.3. We selected these two corpora because both are large, and they represent very different conversational goals. Ubuntu conversations are usually problem solving, while Twitter conversations usually involve casual social interaction or general chit chat.

4.1 Training and Evaluation Procedures

All models are optimized with Adam [13]. Hyperparameters are selected using validation performance, and early stopping with patience is applied based on the variational lower bound [9]. At test time, we apply beam search with a beam width of 5 for decoding the RNN-based generators [10]. For the VHRED models, we sample the latent variable $z_n$at each step and condition on this value when performing beam search. For the Ubuntu dataset we use 300 dimensional word embeddings, and for Twitter we use 400 dimensional embeddings. All models are trained using learning rates of either 0.0001 or 0.0002 with mini-batch sizes of 40 or 80 examples. We apply a truncated version of backpropagation through time and also use gradient clipping. Additional information is available in Appendix 6.2.

Baselines
On both the Twitter and Ubuntu datasets, we include an LSTM baseline with 2000 hidden units.
For Ubuntu, the HRED architecture uses 500 hidden units in the encoder RNN, 1000 in the context RNN, and 500 in the decoder RNN. The encoder uses a standard GRU.
For Twitter, the HRED encoder is a bidirectional GRU where the forward and backward directions each contain 1000 hidden units, and the context RNN and decoder RNN both contain 1000 hidden units.
For comparison, we also include a non neural baseline: the TF-IDF retrieval model described in [17].

VHRED
The encoder and context RNNs in the VHRED follow the same parameter choices as the HRED models. The only change in the decoder is that the context RNN output is concatenated with the sampled latent variable. The feed-forward networks that define the prior and posterior distributions are initialized with weights drawn from a zero mean Gaussian with variance 0.01 and zero biases. In addition, we scale the diagonal covariance of the prior and posterior by 0.1 to stabilize early training, since large variance values can produce noisy gradients for the reconstruction term, which is harmful at the start of optimization.

The VHRED encoder and context RNNs are initialized using the trained parameters of their corresponding HRED models. We also apply two techniques introduced by Bowman et al. [3]. First, we randomly drop decoder input words with a fixed rate of 25 percent. Second, we multiply the KL terms in Equation (4) by a coefficient that begins at zero and gradually increases to one over the first 60,000 batches for Twitter and 75,000

batches for Ubuntu. These strategies significantly stabilize training and encourage the model to use the latent variables. We also experimented with batch normalization for the feed-forward networks but found that it introduced instability without improving the variational lower bound.

Evaluation

Evaluating dialogue responses accurately is challenging [8, 20]. Although word-overlap metrics inspired by machine translation and information retrieval have been used, Liu et al. [16] show that such metrics correlate poorly with human judgments. For this reason, we include human evaluation to compare the models. We also compute automatic metrics and various statistics to help characterize differences in their outputs.

Human evaluation for the Twitter Dialogue Corpus is conducted on Amazon Mechanical Turk (AMT). We do not run AMT studies for Ubuntu because meaningful evaluation usually requires technical knowledge that most AMT users do not have. The evaluation is framed as a set of pairwise comparisons.4 Each participant is shown a dialogue context and two possible system responses, one from each model. They select the response that best fits the given context. If the participant finds both responses equally suitable or cannot understand the conversation, they may choose neither option. For each model pair we run two versions of the experiment.

Table 1 presents the win, loss, and tie percentages for the VHRED model compared with each baseline system, based on the human study conducted on Twitter. The values represent mean preferences along with 90 percent confidence intervals.

	Short Contexts			Long Contexts		
Opponent	Wins	Losses	Ties	Wins	Losses	Ties
VHRED vs LSTM	32:3 _ 2:4	42:5 _ 2:6	25:2 _ 2:3	41:9 _ 2:2	36:8 _ 2:2	21:3 _ 1:9
VHRED vs HRED	42:0 _ 2:8	31:9 _ 2:6	26:2 _ 2:5	41:5 _ 2:8	29:4 _ 2:6	29:1 _ 2:6
VHRED vs TF-IDF	51:6 _ 3:3	17:9 _ 2:5	30:4 _ 3:0	47:9 _ 3:4	11:7 _ 2:2	40:3 _ 3:4

Table 2 provides sample Twitter outputs from the neural models. The exclamation mark token indicates a speaker change.

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!! !!!
whhhhhhyyyy is it suddenly cold in tallahassee today?? this is florida! how can i dream of the beach when my fingers are icey cold?? → sittin on deck in sun looking at lake travis in austin! enjoying day b4 birthday	**VHRED:** the sun is shining!!! **LSTM:** i know!! !!! !!! ! **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 -
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 amrtwoord 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 something
i love katie waissel more than like, anything in life. → i wanna c the show again! i need <number> meet her again! n i want a b-day tweet nxt week doubt that will happen x	**VHRED:** me too. I'm going to have to wait for her to come back. I'm so excited. **LSTM:** i'm sure she will. i'm sure she'll be fine. **HRED:** i'm sure she'll be fine, i'm sure she'll be fine.
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

Example contexts in the study are grouped by length. Long contexts contain at least 80 unique tokens, while short contexts contain at least 20 tokens, even if they are repeated. This division helps evaluate how effectively each model incorporates extended dialogue history into its generated response. Earlier work suggests that hierarchical RNN models are better equipped for longer contexts [24, 26]. Additional screenshots and details are provided in Appendix 6.4.

4.2 Results of Human Evaluation

The human evaluation results in Table 1 show that VHRED is preferred in most of the comparisons. VHRED is chosen significantly more often than both the HRED and TF-IDF baselines in the short and long context settings. VHRED also outperforms the LSTM model in long context conditions. However, in short contexts the LSTM baseline is preferred over VHRED. This result is likely due to the LSTM model producing highly generic responses, as shown in Table 4. Because the LSTM does not incorporate hierarchical structure, it has a shorter effective memory and therefore tends to generate responses that depend mainly on the final part of the previous utterance. Such general or safe responses can be appropriate across many contexts, which increases the chance that human evaluators will judge them as acceptable.

We also note that relying on generic responses can be problematic for dialogue generation, since this often leads to dull and less engaging conversations. The VHRED model takes a different approach by using latent variable sampling, which increases response diversity and allows it to handle longer contexts more effectively. As a result, VHRED produces longer utterances that carry more semantic information compared to the LSTM outputs (see Tables 3 and 4). Although longer replies introduce a higher chance of small errors, which may reduce the preference score for individual examples, we believe that diversity is essential for generating engaging conversations. In the dialogue systems literature, generic responses are typically used as fallback strategies only when the system has no relevant content to provide [25].

Table 3 presents the evaluation results for one-turn and three-turn dialogue generation using the proposed embedding-based metrics.

	Twitter			Ubuntu		
Model	**Average**	**Greedy**	**Extrema**	**Average**	**Greedy**	**Extrema**
			1-turn			
LSTM	0.512	0.389	0.366	0.23	0.169	0.157
HRED	0.501	0.378	0.355	**0.577**	**0.417**	**0.391**
VHRED	**0.533**	**0.396**	**0.38**	0.542	0.384	0.363
			3-turns			
LSTM	0.657	0.561	0.374	0.638	0.456	0.378
HRED	0.646	0.552	0.364	0.742	0.524	0.432
VHRED	**0.689**	**0.583**	**0.391**	**0.777**	**0.536**	**0.448**

VHRED produces responses that are longer and contain more meaningful content than the LSTM model, which tends to output very generic replies. We also observe that VHRED learns to handle smilies, informal expressions, and slang more effectively (as seen in the first example of Table 2), and it can even continue conversations in different languages (as in the fifth example).5 These qualities are not captured by the human evaluation. In addition, VHRED appears more capable of producing imaginative descriptions or small narrative actions compared with the baseline generative models (as illustrated in the third example). In the final example of Table 2, the VHRED output is more creative, although it may be judged as slightly less appropriate due to mild inconsistency with the preceding context, while the LSTM produces a safer and more generic reply. In the next section, we support these observations with quantitative evidence showing that VHRED consistently generates longer responses with higher information content that maintain semantic relatedness with both the context and the ground truth.

4.3 Results of Metric-based Evaluation

To demonstrate that VHRED responses remain more relevant to the dialogue topic and share closer semantic similarity with ground truth responses, we evaluate the models using three word embedding-based similarity metrics.

The **Embedding Average** metric computes the cosine similarity between two vectors formed by averaging the word embeddings in the model response and in the ground truth response [19]. This metric is a standard method for assessing text similarity.

The **Embedding Extrema** metric also embeds each response into a real-valued vector, but instead of averaging, it takes the most extreme value of each dimension across all tokens. The cosine similarity is then calculated between these two vectors.

The **Embedding Greedy** metric provides a more detailed comparison. For each word in the model response, it finds the most similar word in the human response using cosine

similarity over word embeddings. The average similarity across all aligned word pairs is then computed [23]. This metric considers individual word alignments and is therefore more informative for longer replies.

Although these metrics are not strongly correlated with human evaluations of dialogue quality, we interpret them as indicators of topic relevance. Higher metric scores suggest that the content of the model generated response is more semantically aligned with the human response. To ensure reproducibility, we use public Word2Vec embeddings trained on the Google News dataset.6

We compute all three metrics in two configurations: one where each model generates a single utterance (one-turn), and one where each model generates the next three utterances in sequence (three-turns), as shown in Table 3. The results indicate that VHRED aligns more closely with the ground truth topic than either the LSTM or HRED models. The stronger performance of VHRED in the three-turn setting implies that the decoder and context RNN hidden states are better able to maintain on-topic trajectories across multiple utterances. This finding supports our hypothesis that introducing stochastic latent variables helps guide the learning process toward a more balanced representation of short-term and long-term dependencies. We observe a similar pattern when comparing the generated responses directly with the dialogue context, which further confirms this conclusion.

Table 4 reports the information content of generated responses in the one-turn setting. The table includes the average length of each utterance $| U |$, the word-level entropy $H_w = -\sum_{w \in U} \; p(w) \log \; p(w)$, and the utterance-level entropy H_U, which is computed using the maximum likelihood unigram distribution $p$estimated from the training corpus.

	Twitter			Ubuntu		
Model	$\|U\|$	H_w	H_U	$\|U\|$	H_w	H_U
LSTM	11.21	6.75	75.61	4.27	6.50	27.77
HRED	11.64	6.73	78.35	**11.05**	7.53	**83.16**
VHRED	**12.29**	**6.88**	**84.56**	9.22	**7.70**	71.00
Human	20.57	8.10	166.57	18.30	8.90	162.88

We compute the average response length and the average entropy in bits with respect to the maximum likelihood unigram model for all generated responses (see Table 4). The unigram entropy values are calculated on the preprocessed and tokenized datasets. VHRED produces responses with higher per-word entropy on both the Ubuntu and Twitter datasets when compared with the HRED and LSTM models. On Twitter, VHRED also generates longer utterances, which results in responses that contain an average of six additional bits of information compared with the HRED responses. Since real dialogue data contains even more information per word than any of the generative models, a higher entropy value can be viewed as beneficial. VHRED therefore

compares positively to other recently introduced models, many of which produce very low-entropy and generic replies such as OK or I do not know [24, 15].

The higher entropy produced by VHRED indicates that its responses are generally more diverse than those generated by the HRED and LSTM baselines. This also suggests that the hidden state trajectories in the VHRED model explore a larger portion of the representation space, providing further support for our hypothesis that the stochastic latent variable helps the model balance short-term and long-term aspects of generation.

5 Related Work

The use of stochastic latent variables trained by maximizing a variational lower bound is rooted in the Variational Autoencoder (VAE) framework [14, 21]. VAEs have been applied primarily to continuous domains such as image generation [11]. More recent work has extended these ideas to sequence generation. Examples include Variational Recurrent Neural Networks (VRNN) [6], used for speech and handwriting synthesis, and Stochastic Recurrent Networks (STORN) [1], used for music generation and motion capture modeling. Both VRNN and STORN incorporate stochastic latent variables inside RNN architectures, but they draw a separate latent variable at every decoder time step. This design does not capitalize on the hierarchical organization present in many sequential datasets, and therefore does not capture higher-level variability.

Work related to ours includes the Variational Recurrent Autoencoder [7] and the Variational Autoencoder Language Model [3]. These models use encoder-decoder structures for music generation and language modeling respectively. The VHRED model differs from these approaches in several important ways. In VHRED, each latent variable is conditioned on all previously seen sub-sequences, allowing the model to generate multiple sentences while making the latent variables interdependent through the observed tokens. VHRED also builds directly on the hierarchical structure of the HRED model, which makes it suitable for generation conditioned on long contexts. It maintains a deterministic connection between the context RNN and the decoder RNN, enabling the system to share deterministic information between its components.7 Most importantly, VHRED achieves improved performance in tasks that go beyond the typical autoencoder objective. Instead of learning to reconstruct inputs, VHRED is designed for conditional generation of the next utterance in dialogue, which is a significantly more challenging and practical task.

6 Discussion

We presented a new latent variable neural architecture called VHRED. The model employs a hierarchical generation strategy that leverages the structural patterns found in sequential data and is trained using a variational lower bound on the log-likelihood. We applied this architecture to the challenging task of dialogue response generation and showed that it improves on previous approaches across several dimensions, including human-rated response quality. The empirical findings underline the strengths of using a hierarchical generation process when modeling sequences with high entropy. It is also important to highlight that the proposed approach is general in nature. In principle, it can be extended to any sequence generation task that contains hierarchical organization, such as document-level machine translation, prediction of user queries on websites, multi-sentence document summarization, multi-sentence image caption generation, and similar tasks.

References

[1] Bayer, J. and Osendorfer, C. (2014). Learning stochastic recurrent networks. In NIPS Workshop on Advances in Variational Inference.
[2] 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.
[3] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. arXiv:1511.06349.
[4] 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.
[5] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In EMNLP.
[6] Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. (2015). A recurrent latent variable model for sequential data. In NIPS.
[7] Fabius, O. and van Amersfoort, J. R. (2014). Variational recurrent auto-encoders. arXiv:1412.6581.
[8] Galley, M., Brockett, C., Sordoni, A., Ji, Y., Auli, M., Quirk, C., Mitchell, M., Gao, J., and Dolan, B. (2015). deltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. In ACL.
[9] Goodfellow, I., Courville, A., and Bengio, Y. (2015). Deep Learning. MIT Press.
[10] Graves, A. (2012). Sequence transduction with recurrent neural networks. In ICML RLW.
[11] Gregor, K., Danihelka, I., Graves, A., and Wierstra, D. (2015). DRAW: A recurrent neural network for image generation. In ICLR.
[12] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation.
[13] Kingma, D. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In

ICLR.
[14] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In ICLR.
[15] Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2016). A diversity-promoting objective function for neural conversation models. In NAACL.
[16] Liu, C.-W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., and Pineau, J. (2016). How NOT to evaluate your dialogue system. arXiv:1603.08023.
[17] Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The Ubuntu Dialogue Corpus. In SIGDIAL.
[18] Mikolov, T., Karafiát, M., Burget, L., Cernock`y, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In INTERSPEECH.
[19] Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In ACL.
[20] Pietquin, O. and Hastie, H. (2013). A survey on metrics for the evaluation of user simulations. The Knowledge Engineering Review.
[21] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In ICML.
[22] Ritter, A., Cherry, C., and Dolan, W. B. (2011). Data-driven response generation in social media. In EMNLP.
[23] 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 Building Educational Applications Workshop, ACL.

[24] 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, pages 3776–3784.

[25] 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, pages 43–48.

[26] 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.

[27] 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.

[28] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In NIPS, pages 3104–3112.

[29] Young, S., Gasic, M., Thomson, B., and Williams, J. D. (2013). POMDP-based statistical spoken dialog systems: A review. IEEE, 101(5), 1160–1179.