Hierarchical Variational Memory Network for Dialogue Generation

Hongshen Chen*
Data Science Lab, JD.com
chenhongshen@jd.com

Zhaochun Ren*
Data Science Lab, JD.com
renzhaochun@jd.com

Jiliang Tang
Data Science and Engineering Lab
Michigan State University
tangjili@msu.edu

Yihong Eric Zhao
JD.com
ericzhao@jd.com

Dawei Yin†
Data Science Lab, JD.com
yindawei@acm.org

ABSTRACT

Dialogue systems help various real applications interact with humans in an intelligent natural way. In dialogue systems, the task of dialogue generation aims to generate utterances given previous utterances as contexts. Among various spectrums of dialogue generation approaches, end-to-end neural generation models have received an increase of attention. These end-to-end neural generation models are capable of generating natural-sounding sentences with a unified neural encoder-decoder network structure. The end-to-end structure sequentially encodes each word in an input context and generates the response word-by-word deterministically during decoding. However, lack of variation and limited ability in capturing long-term dependencies between utterances still challenge existing approaches. In this paper, we propose a novel *hierarchical variational memory network* (HVMN), by adding the hierarchical structure and the variational memory network into a neural encoder-decoder network. By emulating human-to-human dialogues, our proposed method can capture both the high-level abstract variations and long-term memories during dialogue tracking, which enables the random access of relevant dialogue histories. Extensive experiments conducted on three large real-world datasets verify a significant improvement of our proposed model against state-of-the-art baselines for dialogue generation.

CCS CONCEPTS

• **Information systems** → **Retrieval tasks and goals**; **Question answering**;

KEYWORDS

Dialogue generation, Hierarchical Variational Memory Network, Recurrent Encoder-Decoder Model

*These two authors contributed equally to the paper.
†Corresponding author

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186077

ACM Reference Format:
Hongshen Chen, Zhaochun Ren, Jiliang Tang, Yihong Eric Zhao, and Dawei Yin. 2018. Hierarchical Variational Memory Network for Dialogue Generation. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186077

1 INTRODUCTION

Dialogue generation aims to generate natural-sounding replies automatically to exchange information (e.g., knowledge, sentiments, etc.) and complete a variety of specific tasks in a conversation interaction process [35, 44]. In recent years, automatic dialogue generation has received increasing attention in numerous applications from e-commerce technical support to personal assistant tools [1, 2, 7, 29, 30, 36, 44]. Among all these approaches, end-to-end neural generation models [20, 21, 33, 34, 38, 41] have been proved to be capable in multiple dialogue system applications with promising performance. Most of these end-to-end neural generation models apply encoder-decoder architecture based on recurrent neural network, which directly maps an input context to the output response.

However, challenging problems still exist in current neural models for dialogue generation [6]: (1) Meaningless responses: Given a wide range of contexts, dialogue systems trained via neural generation models are still likely to generate short but pointless responses, such as "ha-ha" and "I don't know". Since these neural generation models are deterministic and shorter responses often have higher likelihood in posterior inference, these models inject no variation during the generation of reply [6]; (2) Memory decay: Vanishing memory is always a challenging problem in the task of dialogue generation. A dialogue session usually contains multiple turns, the longer the conversation moves forward, the harder to capture the long-term memorial dependencies. In particular, the injection of variability exacerbates the difficulty to capture the long-term memory. As a consequence, the response generation for long utterance context is extremely hard, which determines the response generation for long utterance context a rather difficult problem.

In this paper, we aim at developing methods that can tackle challenges of meaningless response and long-term memory dependencies, simultaneously. As an example, consider a dialogue session in Figure 1. Compared with Response 1, although both responses are suitable for the context, Response 2 is more meaningful than its counterpart. With the dialogue session moving forward, modeling the utterance dependencies becomes more and more difficult. As

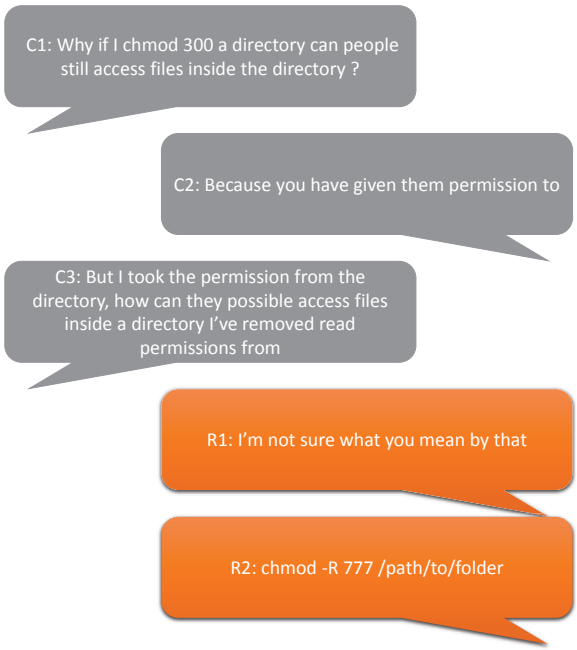

Figure 1: Two different responses for the same context.

a result, the system generates tedious but reasonable responses. Three obstacles hinder the diversification of the response, and long-term dependencies in response generation: (1) utterance sequence modeling is difficult, as a dialogue session consists of multiple utterances; (2) conversation informativeness should be considered not only within an utterance but also across the utterances; and (3) as the conversation goes on, it is important to learn the long-term dependencies.

Our attempt to tackle above obstacles leads to a novel framework, i.e., HVMN, a **H**ierarchical **V**ariational **M**emory **N**etwork (**HVMN**). It combines the spirits of variational autoencoder [18] and memory networks [39, 40] in a hierarchical recurrent neural network setting and aims to memorize the dialogue histories while retrieving the memory with a bit of randomness. The hierarchical structure encodes the utterance sequences, while the variational memory samples a latent variable for each utterance, conditioned on all the previous information, and then memory cells are retrieved by the latent variable. The memory cells keep updated to memorize the new utterance and the response generation decoder is guided by the variational memory block. We conduct extensive experiments on two benchmark datasets and an e-commerce custom-service dataset. Experimental results show that HVMN outperforms state-of-the-art baselines and generates more informative responses.

Our main contributions can be summarized as follows:

- We identify the problems of response informativeness, and long-term utterance dependencies in dialogue generation.
- We build a hierarchical variational memory network model to generate reasonable, informative, and diversified responses by bridging the hierarchical architecture and variational memory network.
- The proposed framework is validated by the extensive experiments and outperforms the state-of-the-arts on both metric-based and human evaluations.

The remaining of the paper is organized as follows. We formulate our research problem in §2 and describe our approach in §3. Then, §4 details our experimental setup and results. Related work is presented in §5. Finally, §6 concludes the paper.

Symbol	Description
D	a dialogue session
U	an utterance
w	a word in an utterance
M	number of utterances in D
m	m'th utterance in D
N	length of an utterance
V	vocabulary
$\mathbf{e}$	embedding mapping function
$\mathbf{h}_t^{enc}$	encoder hidden state at time step t
$\mathbf{h}_t^{dec}$	decoder hidden state at time step t
$\mathbf{h}^u$	hidden state of the utterance encoder
$\mathbf{h}^{con}$	hidden state of the context encoder
$\mathbf{M}$	memory cells
$\mathbf{b}$	variational memory output
$\mathbf{z}$	latent variable
P	prior distribution
Q	posterior distribution
$\mathbf{F}$	forget gate
$\mathbf{U}$	update gate

Table 1: Glossary.

2 PRELIMINARIES

In this section, we first formulate the dialogue generation task formally, and then introduce preliminaries of sequence to sequence models.

2.1 Task Definition

Before getting into the dialogue generation task, We introduce our key notations and concepts. Table 1 lists the main notation we use.

A dialogue session consists of a sequence of M turns of utterances $D = U_1, .., U_M$ between two interlocutors. At m-th turn, given previous utterances $U_1, ..., U_{m-1}$, the dialogue generation model aims to calculate the probability of U_m given $U_1, ..., U_{m-1}$, i.e., $P(U_m|U_1, .., U_{m-1})$.

Each utterance U_m is a variable-length sequence of words, i.e., $w_{m,1}, \dots, w_{m,N_m}$, where $w_{m,n}$ is the nth word of the utterance U_m, and N_m is the length of utterance. $P(U_m|U_1, .., U_{m-1})$ is modeled by decomposing the probability distribution over both the previous utterances and the previously generated words:

$$P(w_{m,1},..., w_{m,N_m}|U_1, ..., U_{m-1}) = \prod_{n=1}^{N_m} P(w_{m,n}|U_1, ..., U_{m-1}, w_{<n}). \quad (1)$$

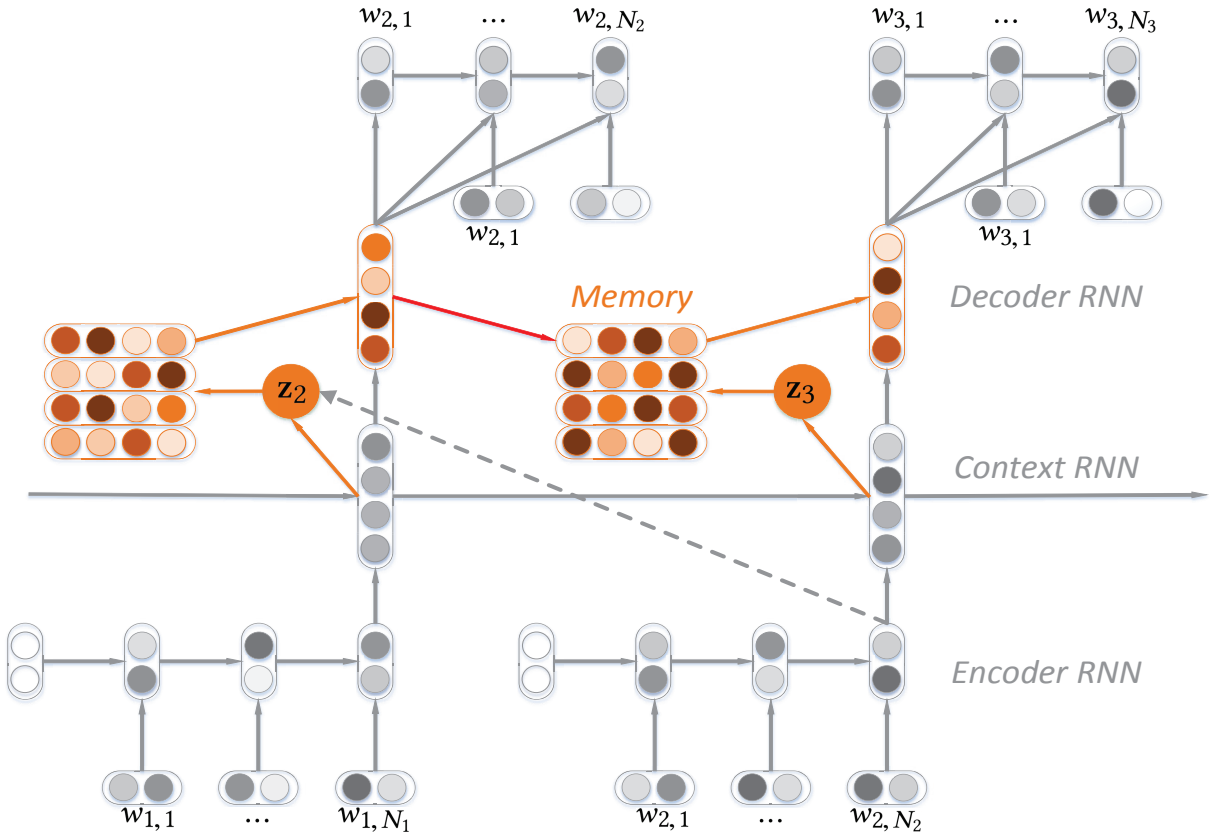

Figure 2: Structure of the hierarchical variational memory network. Orange lines indicate the part of memory reading. Red line is the part of memory updating. Dashed line denotes our posterior approximation.

2.2 Sequence to Sequence Models

Our work is based on sequence-to-sequence (SEQ2SEQ) models. Sequence-to-sequence models prevail in natural language modeling and machine translation tasks [9]. Shang et al. [34], Sordoni et al. [38] and Vinyals and Le [41] introduced it into conversation modeling. It composed of an encoder, which takes in words sequence of previous utterances and outputs a fix-sized context vector which summarizes all previous utterances. It also provides a decoder to generate the next utterance word-by-word based on the context vector and the recurrent hidden state through the prediction over a discrete vocabulary V.

The encoder and decoder process the input sequence and output sequence based on recurrent neural network (RNN). At each step, the encoder RNN unit takes in a word and updates its hidden state:

$$\mathbf{h}_t^{enc} = \sigma(\mathbf{h}_{t-1}^{enc}, \mathbf{e}_{w_t}), \tag{2}$$

while decoder RNN is computed by

$$\mathbf{h}_t^{dec} = \sigma(\mathbf{h}_{t-1}^{dec}, \mathbf{e}_{w_t}, \mathbf{c}), \tag{3}$$

where σ is a nonlinear activation function. It can be as simple as an element-wise logistic sigmoid function and more complex ones like long short-term memory (LSTM)[16] or Gated Recurrent Unit (GRU) [9]. e is a word embedding mapping function. $\mathbf{c}$ is the last hidden state of encoder RNN, which is the summary of the whole input sequence.

The output distribution of the decoder is used to predict the next token and is parameterized by a softmax function over an affine transformation of the decoder RNN hidden state $\mathbf{h}_t^{dec}$:

$$P_\theta(w_{t+1} = v | w_1, ..., w_t) = \frac{\exp(g(\mathbf{h}_t^{dec}, v))}{\Sigma_{v'} \exp(g(\mathbf{h}_t^{dec}, v'))}. \tag{4}$$

, where g is the affine-transformation function. The model parameters are turned by maximizing the log-likelihood over the training instances by stochastic gradient descent.

3 HIERARCHICAL VARIATIONAL MEMORY NETWORK

In this section, we propose a hierarchical variational memory network (HVMN) to model the dialogue generation process. The model employs a hierarchical RNN to model the dialogue utterances in both the utterance level and the context level. In particular, a dialogue consisting of utterance sequence $U_1, ..., U_m$ is modeled as:

$$P(U_1, ..., U_M) = \prod_{m=1}^{M} P(U_m | U_{<m}), \tag{5}$$

As each utterance U_m is a sequence of words $w_{m,1}, ..., w_{m,N_m}$, a dialogue is then formulated as:

$$P(U_1, ..., U_M) = \prod_{m=1}^{M} \prod_{n=1}^{N_m} P(w_{m,n}|U_{<m}, w_{m,<n}). \quad (6)$$

A variational memory network then utilizes a latent variable and a memory block to capture the abstract and concrete details and complex long-term dependencies during the dialogue tracing. When decoding, the previous generated words of response are also given as the input to the decoder. The whole architecture is shown in Figure 2. HVMN learns to generate sequences in four steps:

(1) The utterance *encoder* RNN encodes each turn of utterance word-by-word into a fixed-size vector, which is then sequentially given as input to the *context* RNN.
(2) The *context* RNN computes a hidden state $\mathbf{h}_m^{con}$ for m'th utterance.
(3) Then at *variational memory* block, a latent variable is sampled to retrieve the memory cells as $\mathbf{b}_m$, meanwhile the memory is also updated with $\mathbf{h}_m^{con}$ to accumulatively memorize the current utterance.
(4) The *decoder* RNN takes $\mathbf{h}_m^{con}$ and $\mathbf{b}_m$ as input to generate the response.

Next, we will first introduce the hierarchical utterance encoding process through *encoder* RNN and *context* RNN in §3.1. Then, we detail the variational memory reading and updating mechanism in §3.2. Finally, the response decoding is depicted in §3.3.

3.1 Utterance Encoding

The hierarchical recurrent neural network [37] regards the utterance sequence in two levels: each utterance is composed of a sequence of words, and a dialogue is a sequence of utterances.

Based on such observation, for an utterance $U_m = w_{m,1}, ..., w_{m,N_m}$, the utterance *encoder* RNN maps it to an utterance vector, which is the hidden state after w_{m,N_m} is processed. A word $w_{m,n}$ is encoded by:

$$\mathbf{h}_{m,n}^{u} = \sigma(\mathbf{h}_{m,n-1}^{u}, \mathbf{e}_{w_{m,n}}), \quad (7)$$

where σ is the nonlinear activation function like LSTM or GRU, $\mathbf{e}_{w_{m,n}}$ is the embedding of word $w_{m,n}$, and $\mathbf{h}_{m,n}^{u}$ is the hidden state after processing $w_{m,n}$. $\mathbf{h}_{m,N_m}^{u}$ is the utterance vector $\mathbf{h}_m^{u}$ of utterance U_m, which can be viewed as the summarization of current utterance.

Then, the *context* RNN records the utterance sequence by:

$$\mathbf{h}_m^{con} = \sigma(\mathbf{h}_{m-1}^{con}, \mathbf{h}_m^{u}), \quad (8)$$

where $\mathbf{h}_m^{con}$ is the hidden state of context RNN given utterance vector $\mathbf{h}_m^{u}$, and σ is the same non-linear activation function in Eq.7. $\mathbf{h}_m^{con}$ is the summary of all the observed previous utterances.

3.2 Variational Memory Network

Variational Memory Reading Mechanism.

The model draws $\mathbf{b}_m$ from memory $\mathbf{M}$ using a continuous stochastic latent variable $\mathbf{z}_m$:

$$\mathbf{b}_m = \prod_{i}^{I} \mathbf{M}_m^i \odot \mathbf{z}_m^i, \quad (9)$$

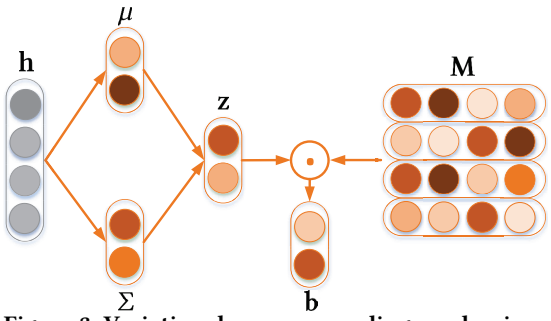

Figure 3: Variational memory reading mechanism.

where $\mathbf{M}_m \in \mathbb{R}^{d_i \times d_z}$, $\mathbf{z}_m \in \mathbb{R}^{d_i}$, and $\odot$ is Hadamard product function. Figure 3 illustrates the variational memory reading mechanism.

The latent variable $\mathbf{z}_m$ is conditioned on all the previous observed tokens, namely the output of *context* RNN $\mathbf{h}_m^{con}$, and is computed for each utterance:

$$P(\mathbf{z}_m|U_{<m}) = N(\mu_{prior}(U_{<m}), \Sigma_{prior}(U_{<m})), \quad (10)$$

where $N(\mu, \Sigma)$ is the multivariate normal distribution with mean $\mu \in \mathbb{R}^{d_i}$ and a constrained diagonal covariance matrix $\Sigma \in \mathbb{R}^{d_i \times d_i}$. $\mathbf{z}_m$ is inferred by maximizing the lower-bound for each utterance:

$$\log P(U_1, ..., U_m) \geq \sum_{m=1}^{M} -KL[Q_\psi(\mathbf{z}_m|U_1, ..., U_m)||P(\mathbf{z}_m|U_{<m})] + \mathbb{E}_{Q_\psi(\mathbf{z}_m|U_1,...,U_m)}[\log P(U_m|\mathbf{z}_m, U_{<m})], \quad (11)$$

where $KL(Q||P)$ is the Kullback-Leibler (KL) divergence between distributions Q and P. The posterior distribution Q_ψ approximates the intractable true posterior distribution:

$$Q_\psi(\mathbf{z}_m|U_1, ..., U_m) = N(\mu_{posterior}(U_1, ..., U_m), \Sigma_{posterior}(U_1, ..., U_m)) \quad (12)$$
$$\approx P_\psi(\mathbf{z}_m|U_1, ..., U_m), \quad (13)$$

where $\mu_{posterior}$ is the approximate posterior mean, and $\Sigma_{posterior}$ is the posterior covariance diagonal matrix conditioned on previous utterances $U_1, ..., U_{m-1}$ and current utterance U_m.

The approximate prior and posterior mean and covariance are computed through a feedforward network, respectively. The former is conditioned on $\mathbf{h}_m^{con}$ and the latter is conditioned on both $\mathbf{h}_m^{con}$ and $\mathbf{h}_{m+1}^{enc}$. A matrix multiplication is performed to compute mean, and a matrix multiplication function followed by a softplus function is implemented to output the diagonal covariance matrix:

$$\mathbf{z}_m^{prior} = \mu_{prior} + \Sigma_{prior} \odot \epsilon_{prior}, \quad (14)$$
$$\mathbf{z}_m^{posterior} = \mu_{posterior} + \Sigma_{posterior} \odot \epsilon_{posterior}, \quad (15)$$

where ϵ_{prior} and $\epsilon_{posterior}$ are standard Gaussian variables.

Variational Memory Updating

Figure 4 illustrates the variational memory updating mechanism. For each utterance, the memory $\mathbf{M}_m$ is updated with $\mathbf{h}_m^{con}$ in order to capture details for long-term dependencies. Inspired by the writing mechanism of neural turing machines [14], we utilize a *Forget* and an *Update* operation. In particular, the forget gate $\mathbf{F}_m \in \mathbb{R}^{d_i}$ defines to what extent the value of each memory cell to be erased, and the update gate $\mathbf{U}_m \in \mathbb{R}^{d_i}$ specializes how much information

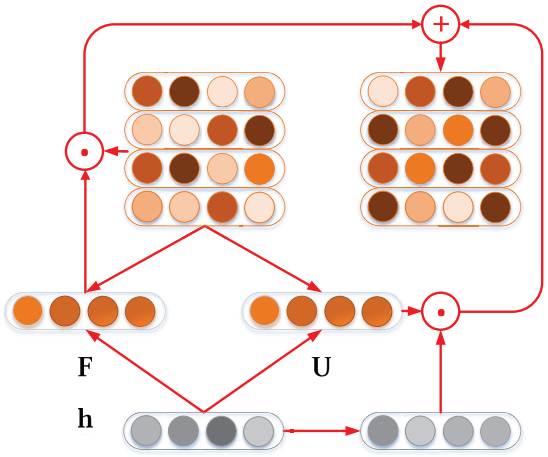

Figure 4: Variational memory updating mechanism. The left memory block is the old memory and the right one is the updated block.

of $\mathbf{h}_m^{con}$ can be written to the memory. Typically, the memory is updated by:

$$\mathbf{M}_m = \mathbf{F}_m \odot \mathbf{M}_{m-1} + \mathbf{U}_m \odot \mathbf{h}_m^{update}, \tag{16}$$

where $\mathbf{F}_m$ and $\mathbf{U}_m$ are computed by a *sigmoid* function parameterized with a linear combination of $\mathbf{h}_m^{con}, \mathbf{b}_m, \mathbf{M}_m$:

$$\mathbf{F}_m = sigmoid(\mathbf{h}_m^{con}, \mathbf{b}_m, \mathbf{M}_m) \tag{17}$$

$$\mathbf{U}_m = sigmoid(\mathbf{h}_m^{con}, \mathbf{b}_m, \mathbf{M}_m) \tag{18}$$

$\mathbf{h}_m^{update}$ is defined as:

$$\mathbf{h}_m^{update} = \sigma(\mathbf{h}_m^{con}, \mathbf{b}_m), \tag{19}$$

where σ is a non-linear activation function like *sigmoid* or *tanh*.

3.3 Dialogue Decoding

During dialogue generation, given the observed utterances, the latent variable $\mathbf{z}_m$ is drawn from the prior $N(\mu, \Sigma)$. Then, $\mathbf{z}_m$ retrieves $\mathbf{b}_m$ from the memory cells. Finally, the embedding of last predicted word $\mathbf{e}_{w_{t-1}}$, $\mathbf{b}_m$, and the output of context RNN $\mathbf{h}_m^{con}$ are given as input to the decoder RNN. In particular, the recurrent unit of decoder is defined as:

$$\mathbf{h}_t^{dec} = \sigma(\mathbf{h}_{t-1}^{dec}, \mathbf{e}_{w_{t-1}}, \mathbf{h}_m^{con}, \mathbf{b}_m), \tag{20}$$

where σ can be a simple *sigmoid* function or GRU, LSTM.

The probability distribution of the next token is given by a softmax function over $\mathbf{h}_t^{dec}$:

$$P_\theta(w_{t+1} = v|\mathbf{h}_t^{dec}) = \frac{\exp(g(\mathbf{h}_t^{dec}, v))}{\Sigma_{v'} \exp(g(\mathbf{h}_t^{dec}, v'))}. \tag{21}$$

, where g is the affine-transformation function.

3.4 Discussions

The conditional prior distribution over variable $\mathbf{z}_m$ injects variation for dialogue generation at the utterance-level, while the conditional distribution over word tokens performs variation at the word-level. As $\mathbf{z}_m$ is constrained by the KL divergence between the prior and posterior, $\mathbf{z}_m$ varies slowly along the utterances, and makes higher-level decisions about what to generate, like the conversation topic, speaker goals or sentiment of the utterance, which helps model long-term output trajectories [32]. In Serban et al. [32], $\mathbf{z}_m$ directly guides the decoder, which increases the response diversity, however, the response appropriateness is weakened due to the lack of long-term memories. As $\mathbf{z}_m$ is a single vector conditioned on all previous observed tokens and injected with Gaussian noise, details of the utterances are unable to be well managed.

In HVMN, the latent variable $\mathbf{z}_m$ incorporates with the memory cells, which mimics the random access of relevant histories, where $\mathbf{z}_m$ focuses on higher-level abstraction, like topics, sentiments, and personalites[22], in the meantime the memory cells specialize in maintaining the long-term details of the observed utterances.Memory cells are retrieved by the stochastic latent variable $\mathbf{z}_m$, and then updated deterministically with the new utterance.

4 EXPERIMENTS

To evaluate the effectiveness of the hierarchical variational memory neural model, we list the following questions to guide the reminder of our experiments:

- **RQ1**: Is our model effective for generating dialogues? Dose it outperforms state-of-the-art baselines?
- **RQ2**: How does our proposed method perform in human evaluation experiments?
- **RQ3**: What is the effect of context length for generating dialogues in our method?

Next, we introduce the datasets in §4.1. The baselines are listed in §4.2 and evaluation metrics are described in §4.3. Details of the training setting are described in §4.4.

4.1 Datasets

In order to assess the performance of our methods, we conduct experiments on three datasets with different styles. Two of them have been used in previous work [26, 43], and another one is extracted from JD.com [1]. Table 2 provides descriptive statistics about our datasets.

Ubuntu Technical Corpus

Our first dataset, the *Ubuntu Dialogue Corpus* [26], is an English multi-turn dialogue corpus, containing about 500,000 dialogues extracted from the Ubuntu Internet Relayed Chat channel. A conversation begins with an Ubuntu-related technical problem, and follows by the responses to the questions. We use the preprocessed dataset. The corpus consists of 448,833, 19,584, 18,920 dialogues of training, validation, testing, respectively. The corpus is a large and typically goal driven dataset.

Douban Conversation Corpus

Different from the domain specific Ubuntu corpus, our second dataset, the *Douban Conversation Corpus* [43], is a Chinese multi-turn open domain conversation corpus collected from Douban groups [2], a popular social networking service in China. There exist multiple responses for one context. The corpus contains 0.5 million dialogues for training, more than 20,000 dialogues for validation, and 5,001 dialogues for test.

[1] https://www.jd.com
[2] https://www.douban.com/group

Corpus	#Train	#Validation	#Test	#Avg. Turns	#Avg. Utterances	#Avg. Words	#Vocab
Ubuntu	448833	19584	18920	4.94	7.48	102.21	268487
Douban	501186	20000	5001	7.69	-	130.66	304988
JD	415000	15000	5005	11.83	20.43	266.09	600739

Table 2: Corpus statistics including number of dialogues in training, validation and test sets, average number of turns, utterances, words per dialogue, and vocabulary size.

JD Customer Service Corpus

We collect customer service dialogues from JD.com, named the *JD Customer Service Corpus*. The JD customer service corpus consists of online retailing customer service dialogues [3]. In JD corpus, each conversation is between a customer and a customer service staff. The corpus contains 415,000 dialogues for training, 1,5000 dialogues for validation, and 5,005 for the test. The average number of turns and utterances in a dialogue session are much larger than the above two corpus.

4.2 Baselines

To evaluate the effectiveness of **HVMN**, we make comparisons between the proposed HVMN and the following state-of-the-art neural dialogue generation models in our experiments:

- **SEQ2SEQ**: It is a sequence-to-sequence dialogue model, also known as the recurrent encoder-decoder model [34, 41]. As a widely-used neural machine translation approach, SEQ2SEQ has been successfully applied to dialogue generation [41].
- **HRED**: It is a hierarchical encoder-decoder model for dialogue generation [33].
- **VHRED**: It is a latent variable hierarchical recurrent encoder-decoder model [32].

4.3 Evaluation Methods

We divide our evaluation metrics into automatic evaluation metrics and human evaluation metrics.

Automatic Evaluation Metrics

Evaluating dialogue system is not a trivial problem [25]. Liu et al. [25] disclosed that word-overlap automatic metrics like BLEU [28] or ROUGE[24] are not well correlate with human evaluations regarding response quality. To evaluate the semantic relevance between the candidate response and target response, we adopt three embedding-based topic similarity metrics proposed by Liu et al. [25]: *Embedding Average* (Average), *Embedding Extrema* (Extrema) and *Embedding Greedy* (Greedy) [11, 27, 31]. The embedding-based metrics actually calculate the similarity between the generated response and the actual response in the embedding space, which are alternatives to word-overlap based metrics and actually take the meaning of each word into consideration.

We use the publicly available Word2Vec[4] to train word embedding. For English, we train word embeddings on Google News Corpus, while for Chinese, the word embeddings are approximated on Chinese Giga-word corpus [5] [13], segmented by *zpar*[6] [45]. Sentence-level embedding is approximated by aggregating the individual embeddings of words in the sentence. With the sentence-level embedding, the candidate response and target response can be measured by standard similarity metrics, e.g. cosine similarity.

To evaluate the informativeness of the response (contrast with the general dull and 'safe' responses), we propose an average trigram word entropy metric. In particular, for a word w_n in a response U, the trigram word entropy is defined as:

$$H(w_n) = -p(w_n|w_{n-2}, w_{n-1}) \log(p(w_n|w_{n-2}, w_{n-1})). \quad (22)$$

The trigram model is trained on the training set of each corpus.

Human Evaluation

To further validate the effectiveness of our model, we compare the responses from different models on Ubuntu corpus by human evaluations. We choose Ubuntu corpus because it is a large public available domain specific technical dataset. It is much more convenient to discriminate a better response regarding whether a technical problem is well understood and tackled. For each sequence of utterances, the responses generated by all the systems are listed in a random order. The system IDs are also anonymized. Each dialogue sequence is allocated to an evaluator randomly. Evaluators are requested to choose the responses on two dimensions: appropriateness to the context, and informativeness. They first choose a response that is more appropriate than other responses given the previous dialogue histories fro a dialogue session. Then, they select a response that is more informative and useful than other responses. By utilizing the two dimensions, we hope to discriminate whether a response is reasonable for a dialogue session and whether it is informative enough compared with a generic safe dull response. If the evaluators disagree all listed responses, or if they cannot understand the dialogue context, they can make no choice and skip.

4.4 Training Procedures

All the models are optimized using Adam [17] with a learning rate of 0.0002 and a batch size of 80. The hyperparameters and early stop patience are chosen according to the variational lower-bound. During testing, we apply beam search with five beams to generate responses. For VHRED, we set the dimensions of the latent variable $d_i = 100$. For HVMN, a memory block $\mathbf{M} \in \mathbb{R}^{d_i \times d_z}$ is augmented with $\mathbf{z}_m$, d_i is set to 10 and d_z to 100. The baseline SEQ2SEQ employs LSTM as the recurrent unit with 1000 hidden units, while other models utilize 500 hidden units, and the dimensionality of other parameters are set accordingly. We set the vocabulary size of Ubuntu corpus as 20000; while we set the vocabulary size of both Douban and JD as 50000. For all baselines and HVMN, we apply the truncated back-propagation and the gradient clipping.

In the following subsections, we first look at the overall model performance on metric-based evaluations. Then, the human-based

[3] We release the corpus at https://github.com/chenhongshen/HVMN.
[4] https://code.google.com/archive/p/word2vec/
[5] We use version 5 in our work.
[6] https://github.com/SUTDNLP/ZPar

Model	Average	Greedy	Extrema	H(w)
Ubuntu				
SEQ2SEQ	0.215603	0.168833	0.126480	0.2638
HRED	0.541548	0.411681	0.319299	**0.3082**
VHRED	0.534103	0.402670	0.306242	0.2878
HVMN	**0.558392***	**0.422914***	**0.322032**	0.3002
Douban				
SEQ2SEQ	0.024255	0.002961	0.023805	1.2253
HRED	0.030904	0.003817	0.029889	1.5116
VHRED	0.042774	0.005147	0.041703	1.3671
HVMN	**0.053293**	**0.006507**	**0.051560**	**3.1042**
JD				
SEQ2SEQ	0.309752	0.204973	0.279654	0.3219
HRED	0.737606	0.500789	0.675900	0.3286
VHRED	0.609605	0.413422	0.558891	0.3473
HVMN	**0.752574***	**0.511170***	**0.691818**	**0.3555**

Table 3: Evaluations on embedding-based metrics. "*" denotes significantly better than VHRED with $p \leq 0.01$.

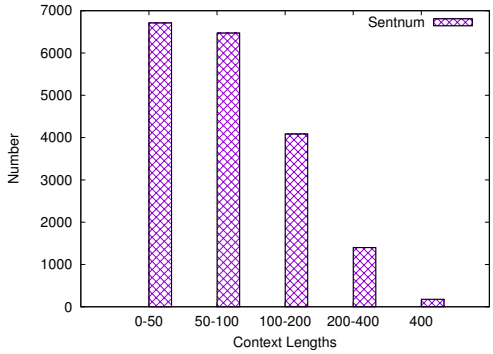

Figure 5: Statistics of number of with respect to different contexts length.

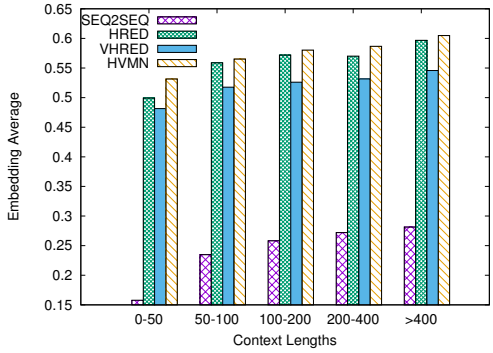

Figure 6: Comparison of embedding average for different length.

evaluations are discussed. Last, we present the response examples for qualitatively evaluation.

4.5 Metric-based Evaluation Results

Table 3 lists the performance of each model. HVMN consistently outperforms almost all the baselines on three corpora, in terms of

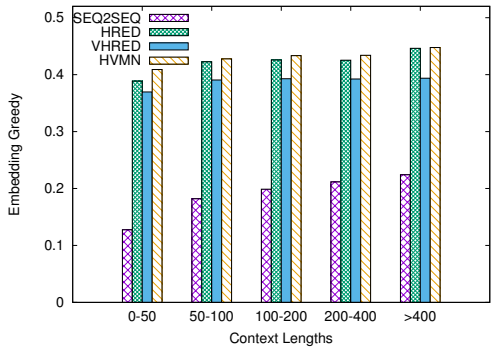

Figure 7: Comparison of embedding greedy for different length.

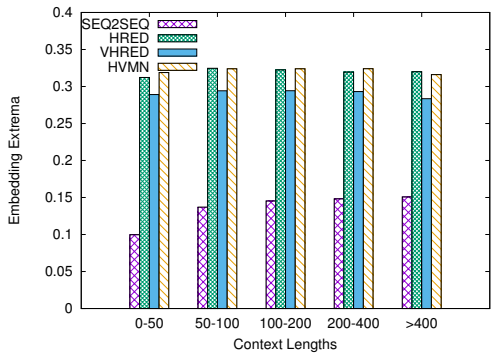

Figure 8: Comparison of embedding extrema for different length.

both the embedding-based topic-similarity metrics and the average trigram word entropy metric (except the word entropy metric on Ubuntu technical corpus). The improvements on Douban dataset are statistically significant (t-test with $p \leq 0.01$) on embedding average and greedy. Not surprised, the performance of SEQ2SEQ is the worst among all methods on all metric-based evaluations. VHRED performs better than HRED on all three topic-similarity metrics on Douban dataset, while HRED actually works better on both Ubuntu and JD dataset. This is consistent with the fact that Douban is an open domain social networking conversation dataset, which is of higher response diversity, while Ubuntu and JD are domain specific dataset. The better performance of VHRED on Douban corpus clearly verifies its ability of increasing the response diversity when comparing with HRED.

For all tested methods, their performance on Douban conversation corpus is much lower than Ubuntu technical corpus and JD e-commerce conversation corpus, while the entropies are much higher. It indicates that the dialogues in Douban corpus are more informative and much harder for dialogue generation task, which agrees with the fact that each context in Douban corpus usually consists of multiple responses. Intuitively, as introduced previously, Douban corpus is an open domain social networking conversation corpus, while Ubuntu technical corpus and JD customer service corpus are more domain specific. The response diversity differs among open domain and domain specific corpus. The higher response entropies on Douban corpus clearly verify such fact.

Comparison	Appropriateness(%)	Informativeness(%)
HVMN *vs* SEQ2SEQ	77.07 : 22.93	84.46 : 15.54
HVMN *vs* HRED	48.52 : 51.48	58.41 : 41.59
VHRED *vs* HRED	44.57 : 55.43	52.91 : 47.09
HVMN *vs* VHRED	53.97 : 46.03	55.56 : 44.44

Table 4: Average differences on human evaluation.

4.6 Effect of Context Length

Figure 5 lists the number of contexts of Ubuntu corpus. Most contexts are of lengths 0-50 and 50-100. Figures 6-8 show the performance on different lengths of contexts for three embedding-based evaluation metrics on Ubuntu corpus. Note that we do not show results in other datasets since we have similar observations. We can see that HVMN performs better than other baselines at most different lengths, while HRED perform the second best. Comparing with HRED, for the contexts with length less than 50, HVMN performs noticeable better on Embedding Average and Embedding Greedy metrics. One possible reason is that, it is harder for short contexts to generate reasonable responses than regular ones, due to the lack of effective information. Through manually examining the results, we observed that baseline models even fail to make meaningful decisions when predicting the response word-by-word, while with the variational memory, HVMN is capable of injecting variation for short contexts. When we looking at the main baseline VHRED, the performance gap between HVMN and VHRED increases when the contexts become longer, which demonstrates that HVMN works better for long contexts than VHRED. It confirms that variational memory in HVMN is able to provide more power for long term dependencies.

4.7 Human Evaluation Performance

To further validate the performance, we conduct the human evaluation on Ubuntu dataset. We choose Ubuntu corpus, because it is easy to specify whether a response is appropriate, and whether a response is informative enough in technical problem discussing.

Table 4 lists the comparison results. Line 1 and line 4 suggest that HVMN outperforms its counterparts, namely 77.07 : 22.93 and 53.97 : 46 : 03 on appropriateness, and 84.46 : 15.54 and 58.41 : 44.44 on informativeness. We also notice that VHRED works better than HRED in terms of informativeness (line 3, 52.91 : 47.09) and performs worse than HRED with respect to appropriateness (44.57 : 55.43). This suggests that, although Ubuntu is a domain specific technical dataset, VHRED still improves response informativeness with a bit loss of appropriateness and HRED tends to make generic safe and meaningless responses. A straightforward observation is that a more informative and longer response tends to make more mistakes comparing with a short and dull response. Therefore, the appropriateness of VHRED decreases compared with HRED, while the informativeness increases a lot.

In line 2, HVMN achieves comparable performance on appropriateness compared with HRED (48.52 : 51.48). However, HVMN remarkably outperforms HRED with respect to informativeness (58.41 : 41.59). This suggests that HVMN maintains its appropriateness while increasing the informativeness. By utilizing the variational memory, HVMN performs better at generate a diversified and "right" response in contrast with VHRED.

4.8 Qualitative Evaluation

Table 5 presents examples of generated responses for different models. It is clear that HVMN performs better at understanding contexts. The responses of HVMN are not only more appropriate but also more informative and useful.

Comparing LSTM and HRED, both systems tend to make generic, "safe" and dull response. LSTM makes short and meaningless responses for all the cases, while HRED outputs longer responses and performs slightly better and is more likely to admit that it does not understand the discussion and fails to give detailed solutions. Comparing HRED and VHRED, line 3 and 5 in Table 5 indicate that VHRED tends to output more specific answers. It gives an appropriate linux command for line 3, however, it fails to generate a proper and effective response for line 5. Generally, HVMN doesn't make a generic "yes/no" response but provides detailed and appropriate operations, especially for line 1 and line 4.

5 RELATED WORK

Our related work can be classified into two categories: variational neural model and memory network.

5.1 Variational Neural Model

Kingma and Welling [18] and Kingma et al. [19] introduce variational neural networks to perform efficient inference and learning in directed probabilities models on a large-scale dataset. They approximate the posterior by a neural inference model. The model parameters are optimized jointly with a reparameterized variational lower bound using the standard stochastic gradient descent. Bayer and Osendorfer [3] propose stochastic recurrent networks for music generation and motion capture modeling. Chung et al. [10] incorporate latent variables into the hidden state of a recurrent neural network. They use the latent variables in token level and apply for speech and handwriting synthesis. Gregor et al. [15] use it in images generation, which combines a novel spatial attention mechanism that mimics the foveation of human eyes, with a sequential variational auto-encoding framework that allows the iterative construction of complex images. Bowman et al. [5] develop a variational auto-encoder for unsupervised generative language modeling. Cao and Clark [6] tackle the boring output issue of deterministic dialogue models by introducing a latent variable model for one-shot dialogue response. Serban et al. [32] directly utilize the latent variable at the sub-sequence level in a hierarchical setting. Li et al. [23] employ a variational auto-encoder to describe the observed sentences and the corresponding latent semantic representations for multi-document summarization.

Unlike previous models, which directly utilize the latent variable to guide the generation, the latent variable in our model is used as a key to read from the memory block and learn to make abstract high-level decisions during dialogue tracing.

5.2 Memory Network

Memory Network model is first proposed by Sukhbaatar et al. [40] and Sukhbaatar et al. [39]. Cheng et al. [8] equip the machine reader with a memory tape, which enables the model to read all the previous hidden state directly. Wang et al. [42] use an interactive memory

Contexts	Response
hello , I'am using ubunutu 12.04 and have black screen It seems that I have graphical system X loggeed as root user , but not as my user . Can someone to end up my problem ? I have ubuntu 12.04 → are you logged on ? and do you have a text prompt ?	**HVMN**: yes , I can log in as root , but I am not able to log in as root **LSTM**: **unknown** **HRED**: yes **VHRED**: yes
hey what command would I use to extract a gz file tar . gz → in which part ? document , assignment , labs ? → sorry labs → wait a second, my connection gets slow connection → oh sorry , can you not see it ? do you know if I have to extract that . tar . gz → not yet finish :D skydrive was suck → lol its the only option I had , i could just send you my onenote linke → okey just for your link I get slower connections :D	**HVMN**: I am trying to get it to work , but I can't find it anywhere **LSTM**: ? **HRED**: yeah I know , but I have to go to bed , I have to go to bed :) **VHRED**: ok
I've been told that du doesn't count hidden files/directories size toward their parent directory size , yet in practice it seems du still does it → well read the manual . because it always did and always will ?	**HVMN**: that 's what I thought , but I don't think it 's possible to do that **LSTM**: ? **HRED**: I know , but I don't know how to do that , I just want to know if there is a way to do it **VHRED**: df -h returns nothing ?
Hello . I want to access a ubuntu desktop from windows 7 to know what the client is doing . How can i do that ? Hello . I want to access a ubuntu desktop from windows 7 to know what the client is doing . How can i do that ? → Try Teamviewer → I want to do without the knowledge of client	**HVMN** You'll need to set up a VNC server on the Ubuntu machine . **LSTM**: ? **HRED**: Sorry , I don't know how to do that . **VHRED**: Then I don't know , sorry .
Hi there , i'm doing an rsync on a server , ive just looked at it remotely and its showing as 3 occurrences of it running(via ps aux | grep rsync) is that normal ? → Do they all look exactly the same or is there any destingueal be difference → all look exactly the same : s whats that mean then ?	**HVMN** The rsync command will tell you if the rsync daemon is running . **LSTM**: **unknown** **HRED**: I don't know what you mean by " **unknown** " **VHRED**: grep " **unknown** . **unknown** "

Table 5: Examples of dialogues generated by various models on Ubuntu corpus. Column 1 lists the context of a dialogue, whereas Column 2 lists the generated response by different models. "→" denotes a turn exchange.

to enhance the long distance memory ability of the decoder in neural machine translation. Bordes and Weston [4] employ memory networks to handle restaurant reservations, using a small number of keywords to handle entity types in a knowledge base (cuisine type, location, price range, party size, rating, phone number, and address). Ghazvininejad et al. [12] adapt it to memorize the relevant grounded facts for a neural conversation model.

However, all these models read the memories in somewhat a deterministic way, while we inject the variability for memory reading through the latent variable. Another difference is that the memory is augmented in a hierarchical setting, and is updated to memorize each utterance.

6 CONCLUSION

In this paper, we have studied the dialogue generation problem and have identified the main challenges: the long-term dependency and the informativeness. To tackle these problems, we have utilized the hierarchical structure, together with the variational memory to enhance the utterance modeling for dialogues generation. The two-level hierarchical structure naturally encodes the dialogue utterances within an utterance and across the utterance sequence. The variational memory tracks the high level abstraction, memorize the long-term details of the observed utterances, and randomly access the dialogue histories. Extensive experiments conducted on two benchmark datasets and a real-world e-commerce dataset have verified the effectiveness of our proposed method by showing significant improvements over multiple baselines in terms of metric-based evaluations and human evaluations.

Note that, our model is not only limited to dialogue generation task, it can also be applied to other tasks like machine reading and summarization. We would like to make further studies in our future work. Also, we would like to integrate external knowledge base into the memory block to make the dialogue generation capable of handling enormous structured knowledge.

ACKNOWLEDGEMENTS

The authors wish to thank the anonymous reviewers for their helpful comments. The authors wish to thank Xi Xiong for preprocessing the JD customer service corpus. Jiliang Tang is supported by the National Science Foundation (NSF) under grant number IIS-1714741 and IIS-1715940.

REFERENCES

[1] D. Ameixa, L. Coheur, P. Fialho, and P. Quaresma. *Luke, I am Your Father: Dealing with Out-of-Domain Requests by Using Movies Subtitles.* Springer International Publishing, 2014.

[2] R. E. Banchs and H. Li. Iris: a chat-oriented dialogue system based on the vector space model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 37–42, 2013.

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

[4] A. Bordes and J. Weston. Learning end-to-end goal-oriented dialog. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[5] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *Proceedings of 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2015.

[6] K. Cao and S. Clark. Latent variable dialogue models and their diversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 182–187, 2017.

[7] H. Chen, X. Liu, D. Yin, and J. Tang. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*, 19(2), 2017.

[8] J. Cheng, L. Dong, and M. Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, 2016.

[9] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[10] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2980–2988, 2015.

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

[12] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-t. Yih, and M. Galley. A knowledge-grounded neural conversation model. *arXiv preprint arXiv:1702.01932*, 2017.

[13] D. Graff and K. Chen. Chinese gigaword. *LDC Catalog No.: LDC2003T09, ISBN*, 1: 58563–58230, 2005.

[14] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[15] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pages 1462–1471, 2015.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

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

[19] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. *Advances in Neural Information Processing Systems*, 4:3581–3589, 2014.

[20] J. Li, M. Galley, C. Brockett, J. Gao, and B. 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*, pages 110–119, 2016.

[21] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and B. Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 994–1003, 2016.

[22] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and B. Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 994–1003, 2016.

[23] P. Li, Z. Wang, W. Lam, Z. Ren, and L. Bing. Salience estimation via variational auto-encoders for multi-document summarization. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 3497–3503, 2017.

[24] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In S. S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[25] C. W. Liu, R. Lowe, I. Serban, M. Noseworthy, L. Charlin, and J. Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, 2016.

[26] R. Lowe, N. Pow, I. Serban, and J. Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, 2015.

[27] J. Mitchell and M. Lapata. Vector-based models of semantic composition. In *Proceedings of The 46th Annual Meeting of the Association for Computational Linguistics*, pages 236–244, 2008.

[28] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Meeting on Association for Computational Linguistics*, pages 311–318, 2002.

[29] Z. Ren, H. Song, P. Li, S. Liang, J. Ma, and M. de Rijke. Using sparse coding for answer summarization in non-factoid community question-answering. In *SIGIR Workshop: Web Question Answering, Beyond Factoids*, 2016.

[30] A. Ritter, C. Cherry, and W. B. Dolan. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 583–593, 2011.

[31] V. Rus and M. Lintean. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 157–162, 2012.

[32] I. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2017.

[33] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 3776–3784, 2016.

[34] L. Shang, Z. Lu, and H. Li. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1577–1586, 2015.

[35] B. A. Shawar and E. Atwell. Chatbots: are they really useful? *Ldv Forum*, 22(1): 29–49, 2007.

[36] H. Song, Z. Ren, S. Liang, P. Li, J. Ma, and M. de Rijke. Summarizing answers in non-factoid community question-answering. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, pages 405–414, 2017.

[37] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562, 2015.

[38] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 196–205, 2015.

[39] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[40] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

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

[42] M. Wang, Z. Lu, H. Li, and Q. Liu. Memory-enhanced decoder for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 278–286, Austin, Texas, November 2016. Association for Computational Linguistics.

[43] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 496–505, 2017.

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

[45] Y. Zhang and S. Clark. Syntactic processing using the generalized perceptron and beam search. *Computational linguistics*, 37(1):105–151, 2011.