Variational Hierarchical User-based Conversation Model

JinYeong Bak
School of Computing
KAIST
jy.bak@kaist.ac.kr

Alice Oh
School of Computing
KAIST
alice.oh@kaist.edu

Abstract

Generating appropriate conversation responses requires careful modeling of the utterances and speakers together. Some recent approaches to response generation model both the utterances and the speakers, but these approaches tend to generate responses that are overly tailored to the speakers. To overcome this limitation, we propose a new model with a stochastic variable designed to capture the speaker information and deliver it to the conversational context. An important part of this model is the network of speakers in which each speaker is connected to one or more conversational partner, and this network is then used to model the speakers better. To test whether our model generates more appropriate conversation responses, we build a new conversation corpus containing approximately 27,000 speakers and 770,000 conversations. With this corpus, we run experiments of generating conversational responses and compare our model with other state-of-the-art models. By automatic evaluation metrics and human evaluation, we show that our model outperforms other models in generating appropriate responses. An additional advantage of our model is that it generates better responses for various new user scenarios, for example when one of the speakers is a known user in our corpus but the partner is a new user. For replicability, we make available all our code and data[1].

1 Introduction

In conversation response generation, modeling the speakers in addition to the utterances is important for generating appropriate responses. Knowing information about a speaker, such as her linguistic style or personal information can help predict her response, and knowing more about both speakers from their previous conversations can help predict the contents of their conversation. Some recent work models the speakers in addition to the utterances (Li et al., 2016b; Olabiyi et al., 2018), but these models tend to overly emphasize the speaker such that they generate very similar responses even when the previous utterances are different. Another difficult and under-tackled problem in conversation response generation is the cold start problem, when the training data do not contain one or both of the speakers. It then becomes very difficult to predict the appropriate responses.

In this paper, we propose Variational Hierarchical User-based Conversation Model (VHUCM) which has a stochastic variable conditioned on the speakers and affects the context. We generate the stochastic variable from a prior distribution whose parameters are given by feed-forward neural networks. The inputs of the neural networks are two speakers that are represented as vector embeddings. Then, we use an RNN to infer the conversational context from the utterances and the stochastic variable. During training, we sample the variable from the variational distribution whose inputs are the speakers and utterances and minimize the difference between the distributions. The stochastic variable models the speakers to a more appropriate degree than the previous models because it is generated from the learned distribution. With VHUCM, we devise a simple solution to the cold start problem by initializing the embeddings of the new speakers by combining the embeddings of their conversation partners. This is based on the assumption that two speakers are close in the embedding space when they have conversations because people try to minimize the social difference among themselves when they have conversations (Linell et al., 1991).

To evaluate VHUCM, we build a new corpus that better reflects the real life scenario of multiple

[1] https://github.com/NoSyu/VHUCM

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1941–1950, Hong Kong, China, November 3–7, 2019.

speakers interconnected in a social network. This corpus contains naturally-occurring conversations over a long period, each conversation having more than just everyday greetings. More details about the corpus can be found in Section 3.

We evaluate VHUCM with this new corpus, and in comparison with other conversation models, the responses generated by VHUCM score the highest using automatic evaluation metrics as well as human evaluation. We show two additional advantages of VHUCM: 1) it can generate personalized responses based on the user and conversation partner, and 2) VHUCM with user embeddings can solve the new user cold start problem.

Our contributions in this paper include the following. First, we collect a large and longitudinal conversation corpus from Twitter (Sec 3). Second, we present VHUCM, a new conversation model which captures the speaker information more effectively and leverages the network of the speakers for better speaker embedding (Sec 4). Third, we conduct the response generation experiments with VHUCM and other models and show that VHUCM outperforms the others (Sec 5). Last, we show how to approach the new user problem with VHUCM and speaker embedding (Sec 6).

2 Related Work

Dialogue response generation has been extensively studied (Ratnaparkhi, 2002; Ritter et al., 2011), and recently, neural network models, especially sequence-to-sequence models have been widely used (Sordoni et al., 2015; Serban et al., 2017; Park et al., 2018; Du et al., 2018; Gu et al., 2019). One limitation of basic seq2seq models is that they only generate responses to the immediately preceding utterances, whereas people usually respond to the entire dialogue consisting of multiple previous utterances. To overcome this limitation, Hierarchical recurrent encoder-decoder (HRED) (Sordoni et al., 2015) builds one more RNN that models the dependency over the utterances in the conversation. VHUCM also constructs the hierarchical RNN structure to understand the previous utterances.

Recently, latent variable models based on Conditional Variational Auto-Encoder (CVAE) (Kingma et al., 2014) or Generative Adversarial Network (GAN) (Goodfellow et al., 2014) show the better performance for generating response (Serban et al., 2017; Xu et al., 2017; Li et al., 2017a; Park et al., 2018). We adopt CVAE to VHUCM and compare the performance with GAN based model (Gu et al., 2019).

Modeling of speakers in the conversation model has been studied (Li et al., 2016b; Xing and Fernández, 2018). They incorporate the speakers to generate the responses, but Li et al. (2016b) only considers a short context of the conversation. Olabiyi et al. (2018) overcomes this, but the user information is still in the utterance level. This approach tends to generate the same response for the same speaker even when the given utterances are different. This is because it gives too much importance to the speaker rather than the content of the previous utterances. VHUCM differs from these models in that it uses a global stochastic variable which is conditioned on the speakers and affects the context.

Users	Dyads	Conv's	Utterances
27,152	107,611	770,739	6,109,469

Table 1: Basic statistics of Twitter conversations corpus

3 Twitter Conversation Corpus

In this section, we describe our new Twitter conversations corpus. We first explain how we build the corpus, then we compare it with other conversation corpora.

3.1 Definition and Basic Statistics

We define a Twitter conversation as a chain of tweets where two users are consecutively replying to each other's tweets using the Twitter reply button. Unlike other research using Twitter conversation corpora (Bak et al., 2014; Li et al., 2016b), we increase the minimum number of the tweets in a conversation from three to five because in many cases, the first few utterances are greetings such as "How are you". To model the users in-depth, we keep conversations only from dyads with ten or more conversations and users with three or more conversational partners.

Our Twitter conversation corpus consists of 27,152 users, 107,611 dyads, 770,739 conversations and 6,109,469 tweets which were posted between April 2007 to June 2013. The average duration of each conversation is around 4.5 hours, and the average duration between the first and the last conversations of each dyad is around 110 days. Table 1 summarizes the corpus.

3.2 Other Conversation Corpora

Our corpus has two major differences from existing conversation corpora. First major difference is that our corpus consists of open-domain naturally-occurring conversations. The widely-used Cornell movie corpus (Danescu-Niculescu-Mizil and Lee, 2011) and TV series transcripts (Li et al., 2016b) are made up conversations written by the script writers. The Ubuntu dialog corpus (Lowe et al., 2015) consists of naturally-occurring conversations, but the topics are limited to a specific computer OS. Mazare et al. (2018) creates a corpus perhaps closest to open-domain naturally-occurring conversations, but Reddit comments are in the form of discussions, rather than personal casual conversations.

Another property of our corpus is that it is large and longitudinal. It contains ten or more conversations by the same two users, over several months, and the corpus contains hundreds of thousands of such dyads and their conversations. Existing Twitter conversation corpora (Ritter et al., 2011; Li et al., 2016b) consist of short conversations, with about three turns in each conversation. The DailyDialog corpus (Li et al., 2017b) contains daily conversations from English learners, but the corpus size is relatively small, and it does not have a user indicator to identify the speakers. Zhang et al. (2018) builds a dialogue corpus for personalized responses, but the corpus is generated by crowd workers in a controlled setting, and the corpus is relatively small.

4 Variational Hierarchical User-based Conversation Model

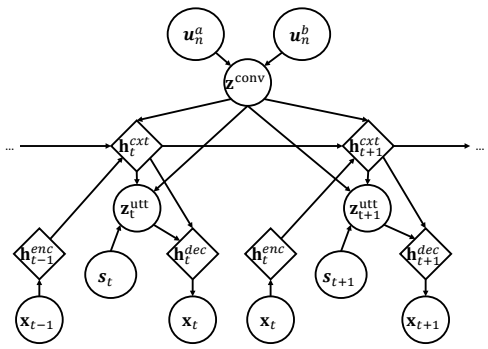


Figure 1: Graphical representation of VHUCM. The global context of the conversation $\mathbf{z}^{conv}$ is inferred by the speakers of the conversation.

This section describes our model, the Variational Hierarchical User-based Conversation Model (VHUCM) which explicitly models the speakers as well as the input utterances. VHUCM is based on the Variational Hierarchical Conversation RNN (VHCR) (Park et al., 2018), but in generating the conversation level latent variable $\mathbf{z}^{conv}$ and the uttrerance level latent variable $\mathbf{z}_t^{utt}$, we incorporate the user embedding. We further propose to initialize the user embedding from the conversation network described in section 4.3.

4.1 Notations

We have N conversations in the data $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N\}$, where the n-th conversation has M_n utterances by two users u_n^a and u_n^b. The t-th utterance in the conversation has word sequence $\mathbf{x}_{nt}$ and speaker indicator s_{nt}. Hence, $\mathbf{c}_n = \{(\mathbf{x}_{n1}, s_{n1}), \ldots, (\mathbf{x}_{nM_n}, s_{nM_n})\}$ and $s_{nt} \in \{u_n^a, u_n^b\}$ where $t \in \{1, \ldots, M_n\}$.

4.2 VHUCM

The structure of VHUCM is similar to the VHCR, which has three RNNs (encoder RNN $f_{\boldsymbol{\theta}}^{enc}$, context RNN $f_{\boldsymbol{\theta}}^{cxt}$, and decoder RNN $f_{\boldsymbol{\theta}}^{dec}$) and two latent random variables ($\mathbf{z}_t^{utt}$ and $\mathbf{z}^{conv}$).

Given previous $t-1$ utterances[2], VHUCM generates the word sequence of the next utterance $\mathbf{x}_t$ as follows:

$$\boldsymbol{h}_{t-1}^{enc} = f_{\boldsymbol{\theta}}^{enc}(\mathbf{x}_{t-1})$$
$$\boldsymbol{h}_t^{cxt} = f_{\boldsymbol{\theta}}^{cxt}(\boldsymbol{h}_{t-1}^{cxt}, \boldsymbol{h}_{t-1}^{enc}, \mathbf{z}^{conv}).$$

We use the embeddings of the two users in the conversation $\boldsymbol{h}_{u^a}^{user}$ and $\boldsymbol{h}_{u^b}^{user}$ to create dyad-specific context $\mathbf{z}^{conv}$ as follows:

$$p_{\boldsymbol{\theta}}(\mathbf{z}^{conv} | \boldsymbol{h}_{u^a}^{user}, \boldsymbol{h}_{u^b}^{user}) = \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}^d, \boldsymbol{\sigma}^d \mathbf{I})$$
$$\boldsymbol{\mu}^d = \text{MLP}_{\boldsymbol{\theta}}(\boldsymbol{h}_{u^a}^{user}, \boldsymbol{h}_{u^b}^{user})$$
$$\boldsymbol{\sigma}^d = \text{Softplus}(\text{MLP}_{\boldsymbol{\theta}}(\boldsymbol{h}_{u^a}^{user}, \boldsymbol{h}_{u^b}^{user})).$$

We also use the user embeddings to understand the users' words as follows:

$$p_{\boldsymbol{\theta}}(\mathbf{z}_t^{utt} | \mathbf{x}_{<t}, \mathbf{z}^{conv}, \boldsymbol{h}_{s_t}^{user}) = \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t \mathbf{I})$$
$$\boldsymbol{\mu}_t = \text{MLP}_{\boldsymbol{\theta}}(\boldsymbol{h}_t^{cxt}, \mathbf{z}^{conv}, \boldsymbol{h}_{s_t}^{user})$$
$$\boldsymbol{\sigma}_t = \text{Softplus}(\text{MLP}_{\boldsymbol{\theta}}(\boldsymbol{h}_t^{cxt}, \mathbf{z}^{conv}, \boldsymbol{h}_{s_t}^{user})).$$

To decode the words from the context, we use the decoder RNN with context-related variables.

$$p_{\boldsymbol{\theta}}(\mathbf{x}_t | \mathbf{x}_{<t}, \mathbf{z}_t^{utt}, \mathbf{z}^{conv}) = f_{\boldsymbol{\theta}}^{dec}(\boldsymbol{h}_t^{cxt}, \mathbf{z}_t^{utt}, \mathbf{z}^{conv})$$

For the inference of $\mathbf{z}^{conv}$, we adopt the idea of VHCR that uses a bi-directional GRU f^{conv} where

[2] In order to be concise, we remove the notation n.

the input is the encoder RNN outputs.

$$q_\phi(\mathbf{z}^{conv}|\mathbf{c}, \boldsymbol{h}^{user}_{u^a}, \boldsymbol{h}^{user}_{u^b}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}^{Id}, \boldsymbol{\sigma}^{Id}\mathbf{I})$$

$$\boldsymbol{h}^{conv} = f^{conv}(\boldsymbol{h}^{enc}_1, \ldots, \boldsymbol{h}^{enc}_N)$$

$$\boldsymbol{\mu}^{Id} = \text{MLP}_\phi(\boldsymbol{h}^{conv}, \boldsymbol{h}^{user}_{u^a}, \boldsymbol{h}^{user}_{u^b})$$

$$\boldsymbol{\sigma}^{Id} = \text{Softplus}(\text{MLP}_\phi(\boldsymbol{h}^{conv}, \boldsymbol{h}^{user}_{u^a}, \boldsymbol{h}^{user}_{u^b}))$$

To infer $\mathbf{z}^{utt}_t$, we build the additional networks:

$$q_\phi(\mathbf{z}^{utt}_t|\mathbf{c}, \mathbf{z}^{conv}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}'_t, \boldsymbol{\sigma}'_t\mathbf{I})$$

$$\boldsymbol{\mu}'_t = \text{MLP}_\phi(\mathbf{x}_t, \boldsymbol{h}^{cxt}_t, \mathbf{z}^{conv}, \boldsymbol{h}^{user}_{s_t})$$

$$\boldsymbol{\sigma}'_t = \text{Softplus}(\text{MLP}_\phi(\mathbf{x}_t, \boldsymbol{h}^{cxt}_t, \mathbf{z}^{conv}, \boldsymbol{h}^{user}_{s_t})).$$

The loss function of VHUCM is the ELBO loss and the auxiliary loss. The ELBO loss is the combination of the reconstruction loss, and the KL divergence between p_θ and q_ϕ with KL cost annealing (Bowman et al., 2016). We also add the bag-of-word loss (Zhao et al., 2017) as the auxiliary loss to avoid the vanishing latent variable problem.

4.3 VHUCM-PUE

We leverage the fact that people connected in a social network can be modeled with the node2vec algorithm (Grover and Leskovec, 2016), where the nodes are users, and two nodes are connected if they have conversations. We can use the number of conversations as the weight of the edge. We extend the original VHUCM with these pre-trained user embeddings based on the conversation network, and we can call this VHUCM-PUE (VHUCM with Pre-trained User Embeddings). This approach is similar to soc2seq (Bhatia et al., 2017) which embeds the users more elaborately using comments and likes but is a much simpler conversational model based on the encoder-decoder.

5 Experiment 1 - Response Quality

This section describes the experiments and results of VHUCM and VHUCM-PUE compared to other state-of-the-art response generation models. We compare the models using various automatic evaluation metrics in section 5.2 and with human evaluation in section 5.3. Section 5.4 presents a qualitative analysis of the generated responses, in particular how the responses can be personalized.

5.1 Experiment Setup

Data We use the corpus described in section 3 for this experiment. We split the data as 80/10/10 for training/validation/test, because VHUCM learns the user information from the conversation, we split the conversations of each dyad in chronological order and merge each part into overall training, validation and test sets. We further investigate cases of new users that are not in the training data but appear in the test data in section 6.

Comparison models We compare VHUCM and VHUCM-PUE with the following models:

- *HRED* (Sordoni et al., 2015): A hierarchical structure of RNNs for encoder, decoder, and context.
- *VHRED* (Serban et al., 2017): A variational autoencoder model that adds latent variables $\mathbf{z}^{utt}_t$ to the HRED.
- *VHCR* (Park et al., 2018): A variational autoencoder model that adds the latent variable $\mathbf{z}^{conv}$ to the VHRED.
- *SpeakAddr* (Li et al., 2016b): A persona seq2seq model. We choose the Speaker-Addressee model as this outperforms the Speaker model.
- *DialogWAE* (Gu et al., 2019): A conditional Wasserstein autoencoder model that uses GAN for training. We use DialogWAE-GMP which outperforms DialogWAE by using the Gaussian mixture prior.

5.2 Quantitative Analysis

Model	Perp	Model	Perp
HRED	72.6	SpeakAddr	85.5
VHRED	71.2	DialogWAE	81.0
VHCR	71.1	VHUCM	65.3
		VHUCM-PUE	**62.7**

Table 2: Perplexity per word of the generated responses. VHUCM-PUE outperforms all other methods.

Metrics To quantitatively compare the response generation performance of VHUCM-PUE with other models, we use various automatic metrics used in (Park et al., 2018; Du et al., 2018; Olabiyi et al., 2018; Gu et al., 2019).

We first compute the perplexity per word of each model using the reconstruction errors of the test data. For variational models, we calculate the values by importance sampling. As shown in Table 2, VHUCM-PUE outperforms the other methods in terms of word perplexity.

Model	BLEU	Embedding			ROUGE-L			Distinct		Len
		Avg	Ext	Gre	Rec	Prec	F	dist-1	dist-2	
HRED	0.090	0.577	0.364	0.357	0.064	0.162	0.066	0.019	0.072	9.4
VHRED	0.120	0.596	0.368	0.377	0.072	0.161	0.072	0.016	0.063	11.4
VHCR	0.137	0.599	0.371	0.381	0.076	0.169	0.075	0.020	0.076	**12.3**
SpeakAddr	0.037	0.567	**0.384**	0.337	0.052	**0.218**	0.055	0.016	0.031	4.8
DialogWAE	0.127	0.586	0.345	0.369	0.079	0.132	0.080	0.012	0.104	11.5
VHUCM	0.120	0.633	0.373	0.394	0.075	0.154	0.079	0.030	0.108	10.1
VHUCM-PUE	**0.161**	**0.643**	0.376	**0.400**	**0.082**	0.162	**0.087**	**0.034**	**0.123**	10.6

Table 3: Comparison of various models using BLEU, Embedding, and ROUGE-L scores, which measure the quality of the generated responses with respect to the ground truth. Embedding Avg, Ext, and Gre are average, extrema and greedy matching by embedding based metrics, respectively. ROUGE-L is ROUGE score using the longest common subsequence. Distinct unigram (dist-1) and bigram (dist-2) measure the degree of diversity among responses. Len is the average length of the generated response. VHUCM-PUE outperforms the other methods for most of the metrics.

We then compute the following metrics, with the results in Table 3:

- *BLEU* (Papineni et al., 2002): We compute the sentence-level BLEU score with the smoothing seven technique (Chen and Cherry, 2014).
- *Embedding based metrics* (Liu et al., 2016): We use three types of the embedding based metrics, *Embedding average*, *Embedding extrema*, and *Embedding greedy matching*. We use pre-trained Google news word embedding (Mikolov et al., 2013) to measure the embedding metrics to avoid dependency between the training data for dialogue response generation and embedding.
- *ROUGE-L* (Lin, 2004): We report three types of ROUGE-L score, *ROUGE-L Rec*, *ROUGE-L Prec*, and *ROUGE-L F* score.
- *Distinct* (Li et al., 2016a): We use *dist-1* and *dist-2* that refer to the diversity of generated responses.
- *Average length*: We examine the average length of the generated responses to show its diversity.

Results As shown in Table 3, VHUCM-PUE outperforms the other methods on most metrics. SpeakAddr generally shows the lowest performance, reflecting that this model does not explicitly model the conversation context. ROUGE-L Prec on SpeakAddr shows higher values than the others because it generates shorter responses on average. VHCR performs better than HRED and VHRED and performs similarly to DialogWAE. VHUCM outperforms VHCR, confirming the effectiveness of the user embeddings. Finally, VHUCM-PUE outperforms all other methods for most of the metrics, showing the effectiveness of the conversation network in initializing the user embeddings.

5.3 Human Evaluation

	Wins	Losses	Ties
vs SpeakAddr	**40.7 ± 3.0**	34.9 ± 3.0	24.4 ± 2.4
vs VHCR	**45.5 ± 2.7**	40.4 ± 2.8	14.1 ± 2.1
vs DialogWAE	**52.5 ± 2.9**	34.5 ± 2.8	13.0 ± 2.1

Table 4: Human evaluation of the appropriateness of the generated response. We ask MTurkers to pick a more appropriate response from two candidates generated by different models. Five annotators answered each task. We compute the mean preferences with a 90% confidence interval. VHUCM-PUE outperforms the baselines.

We also evaluate the generated responses by human judgment using Amazon Mechanical Turk (MTurk). First, we sample 150 dyads randomly and build the tasks. We select three conversations of the dyad from the training data and one conversation from the test data randomly for each task. During the task, we show the three conversations initially, after which we show the three-turn utterances of a dyad from the test data. We show the two candidate responses from different models and ask the Mturk annotators which is more ap-

propriate. One of the candidates is from VHUCM-PUE, and the other is from one of the baselines (SpeakAddr, VHCR, and DialogWAE). The annotators can use the answer 'Tie' if they cannot readily select an answer. For each task, we take five annotations and compute the mean preference with a 90% confidence interval. Table 4 shows the results, in which VHUCM-PUE competes with three baselines and outperforms all of them.

Table 5 shows an example of the generated responses from SpeakAddr, VHCR, DialogWAE, and VHUCM-PUE for the same three-turn context. Overall, VHUCM-PUE generates more appropriate responses given the context. SpeakAddr makes general responses without the context. VHCR and DialogWAE create more context related responses than SpeakAddr. But, VHUCM-PUE generates more sophisticated responses than other baselines.

5.4 Personalized Responses

We test for consistent responses of personal information, such as age when the user is fixed. We devise personal information questions and generate the responses between the dyads. Table 6 shows examples of these questions and responses.

For the question, the answers of "where is your hometown?" from A are identical even when the questioners differ, because A reveals the hometown in the training data. VHUCM-PUE can answer the user-specific context with learned user vector. In the opposite cases when the questioner is A, the responses are based on the answerer.

The answers of A to the question about age are entirely consistent with the experience question. Moreover, the answer of A for question from B is interesting. From the other answers, we know that age of A is 19, and B's age is 18. The answer to the question is correct even it does not generate the exact age number.

To see the relationship between a dyad, we create the third question that asks about the partner, "Do you love me?". The generated responses show that they match the same feeling on each other. For example, the generated response of A & B and A & C dyads are agreed with the question, 'I love you'. The response of A & D dyad are also matched, but negation of the question. We find the reason that A & B and A & C dyads use emotional words and emoticons. But, A & D dyad have question and answers about the computer-related topics perfunctorily. It shows that VHUCM-PUE can learn the relationship of dyads from their conversation word patterns.

Another interesting outcome is that all responses of C contain ':)' and 'xx' words. User C usually uses the words at the end of the tweets in the training data. It also shows that VHUCM-PUE can learn the word preferences of users.

6 Experiment 2 - New Users and Dyads

In this section, we investigate the new user problem when generating responses. We simulate various scenarios of new dyads and new users, and show how VHUCM-PUE incorporates the network information to perform better even for an unknown user.

6.1 Experiment Setup

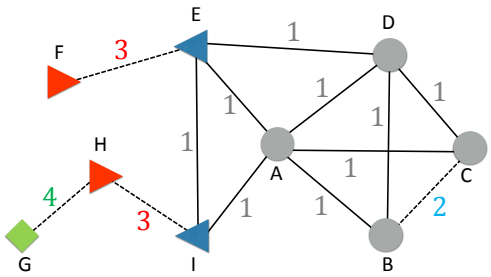


Figure 2: Example of a conversation network. Nodes are users, and a edge between two nodes forms a line if they have conversations. To create the new users and dyads in the test time, we randomly select dyads and remove all conversations between them in the training data (the dashed line in the figure). F, G, and H are new users, B-C dyad is new dyad since they are not in the training data.

We first sort the conversations of a dyad in the order of time, and split the data into training, validation and test data in the same way as in experiment 1. To create new users and dyads, we arbitrarily pick dyads and remove all conversations of those dyads from the training and the validation data. The conversations of these dyads are still in the test data, so we can treat the users in these dyads as new users. Figure 2 shows an example of the conversation network. There are nine users in the figure, and they are connected if they have conversations in the original data. Then, we choose dyads randomly (say B-C, E-F, G-H, and H-I), and remove all conversations of these dyads in the training and validation data. We define four types of dyads with respect to new and known users.

Context	SpeakAddr	VHCR	DialogWAE	VHUCM-PUE
nice ! atl a few months ago and already back again :) how could you not be .. im so excited for the both of them . → whose getting married ? ? → snarky and suzi . got engaged back in dec	i know , i know ! ! !	hey	ah ha ha , i have to do a lot of time to get to see what happens .	what about u
call me when your at the station , i shall be with → am waiting at station 2 caramel macchiato and 2 pain aux raisin in hand ! ! ! x → whoop whoop , , , starting to feel sick	i know , i know ! !	no . i don't even know what to do , but i don't think so . :(	i know what the hell is , but the is .	:(oh .
how was your exam ? x → think it was okay actually :) x → thats good then :) x	you !	:d :d :d :d :d :d :d i miss you too <3 <3 <3	hahah ! ! ! im so sorry to tell you when I get home or something lol	you are still the best person who's there ? x

Table 5: Examples of generated responses from SpeakAddr, VHCR, DialogWAE, and VHUCM-PUE. The → means a changes of turn in the context.

Questioner	Answerer	Questions		
		Where is your hometown ?	How old are you ?	Do you love me ?
User B	User A	north carolina !	i'm not sure , but i am a bit older than you	i love you .
User C	User A	north carolina .	19 ! ! !	yes i do !
User D	User A	north carolina .	i'm 19 . i don't even know what to say	no i do not
User A	User B	minnesota . <unk>.	18 yr old	because i love you .
User A	User C	manchester :) xx	nothing much :)	i love you too :) xx
User A	User D	i live in <unk>.	i have no idea	no , i don't .

Table 6: Responses to users' personal information questions from VHUCM-PUE. The questioner asks each question to the answerer, and VHUCM-PUE generates the answerers' responses. The '<unk>' token is an unknown word. VHUCM-PUE generates personalized responses of users.

1. *Known Dyad*: Conversations of the dyad exist in the training and validation data. Examples include A-B, A-D, and E-I.

2. *Known Users*: There are no conversations of the dyad in the training and validation data, but the two users' conversations with other users exist in the training data. An example is B-C.

3. *Known Partner*: Similar to *Known Users*, except there are no conversations of one of the speakers ("new user") in the training data. But the other partner ("known partner") has conversations with other users in the training corpus, so the model can see the partner's conversations during training. Examples are F-E and H-I where F and H are new users, and E and I are known partners.

4. *New Users*: Both users do not have any conversations in the training and validation data. They are new users. An example is G.

For this experiment, we create a small but dense conversation network from our corpus. As in section 3, we filter out dyads with fewer than ten conversations, but here we filter users who have fewer than five conversation friends. The number of users is 2,187, the number of dyads is 3,833, and the number of conversations is 84,295. We randomly select 20% of the dyads and remove all conversations of the dyads in the training and validation data. We conduct the experiment six times with different random seeds.

6.2 New User Embedding

We assume that two users are close in the user embedding space when they have conversations since people try to minimize the social difference between the others when they have conversations

	Dyads	*NonDyads*
VHUCM	2.449 ± 0.004	2.453 ± 0.001
VHUCM-PUE	2.879 ± 0.003	5.444 ± 0.001

Table 7: Distance between users in the user embedding spaces from VHUCM or VHUCM-PUE. We make two groups of user pairs; pairs who have conversations (*Dyads*), and pairs who do not have any conversations (*NonDyads*). We compute the mean value of the Euclidean distance between each pair in the groups with a 90% confidence interval. VHUCM-PUE distinguishes the two groups significantly better than VHUCM.

(Linell et al., 1991). We investigate how VHUCM and VHUCM-PUE learn this assumption well. We make two groups of user pairs whether they have conversations or not. *Dyads* are user pairs who have conversations, and *NonDyads* are user pairs who do not. And, we compute the Euclidean distance of user embedding between each pair in a group.

Table 7 shows the results. The two groups are statistically significantly different in VHUCM-PUE user embedding, but not in VHUCM. This result shows that VHUCM-PUE learns the assumption better than VHUCM, and this is one reason why VHUCM-PUE outperforms VHUCM.

Accordingly, we can set the vector of a new user to be near the conversation partner. Formally, we compute the average of the conversation partners of the new user and add some random noise to the new user embedding. For example, F is the new user and E is the known partner in Figure 2. We utilize the embedding of F from conversation partner E as follows:

$$\boldsymbol{h}_F^{user} = \sum_{i \in \text{friends of F}} \boldsymbol{h}_i^{user} + \epsilon$$

$$\epsilon \sim \text{Uniform}(u_{min}, u_{max})$$

where u_{min} and u_{max} are min and max values over each dimension of the trained user vectors, and the friends of F $= \{E\}$. We incorporate this method in VHUCM-PUE for new users in the *Known Partner* scenario.

In the *New Users* case, we cannot get the user information during the test. So, we set their embeddings as random values. For example, G is a new user in the *New Users*, thus we set the user embedding values from the uniform distribution over the trained user embedding space as $\boldsymbol{h}_G^{user} \sim \text{Uniform}(u_{min}, u_{max})$. We incorporate this method in VHUCM for all new users and VHUCM-PUE for users who are in the *New Users*.

6.3 Results and Discussion

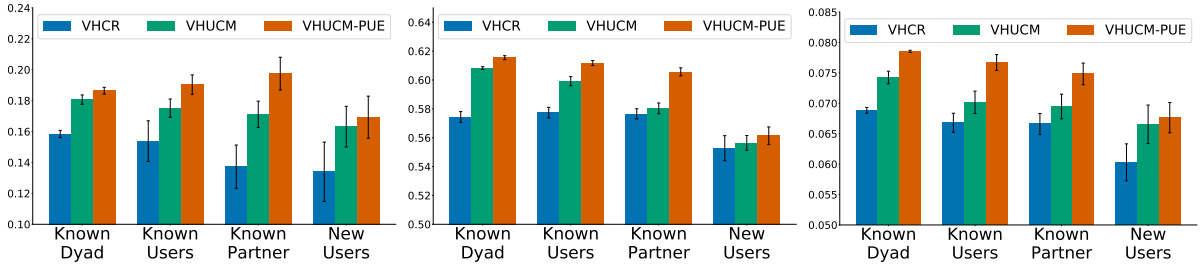
Figure 3 shows the response quality test results for all cases. VHUCM and VHUCM-PUE outperform VHCR in the *Known Dyad* case which is the same situation in section 5. In the *Known Users* case, VHUCM and VHUCM-PUE performs better than VHCR. Although the conversations of the dyad are not in the training data, VHUCM can learn the users from conversations with other partners, and this helps to infer the responses between them.

In the *Known Partner* case, VHUCM-PUE outperforms VHUCM and VHCR. To see the reason of this improvement, we investigate the difference of user embeddings between the models. We run VHUCM-PUE with full data which are not removed the conversations in the training data. Figure 4a shows the two-dimensional plot of the user embedding by projecting t-SNE (Maaten and Hinton, 2008). The new user (▶) and known partner (◀) pairs are closed each other. But the closeness between the dyad is not observed in VHUCM with removed data since it creates the new user embedding randomly (Fig 4b). The new user and known partner dyads are closed in VHUCM-PUE which applies the method in section 6.2 to new users in the space (Fig 4c). This results show the reason why VHUCM-PUE has a better performance in *Known Partner* case.

Finally, *New Users* shows an overlap with regard to the performances of all models. No information about new users exists in the training data and test data; hence VHUCM-PUE is forced to initialize the new users randomly, similarly to VHUCM.

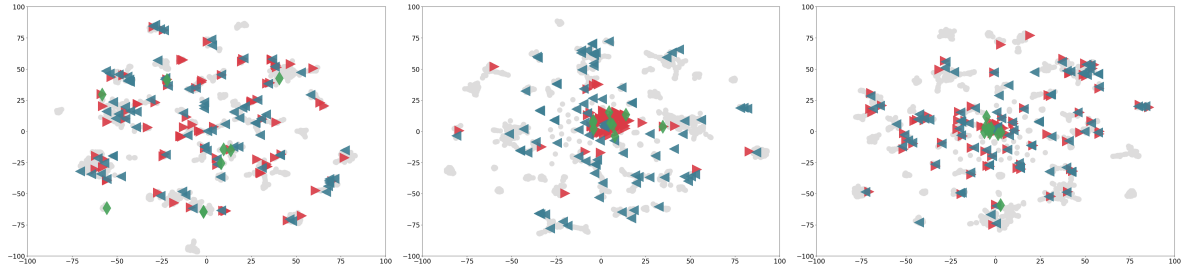
7 Conclusion

In this paper, we made large long-term conversation corpus from Twitter. We have presented VHUCM to generate the response given the prior utterances of a conversation and user. To initialize the users to VHUCM, we pre-train the user embedding from the conversation network. We showed that VHUCM-PUE outperforms others in most metrics. We also suggested the way to incorporate new users who have conversations in test time with trained users. We showed that using learned partners' embedding helps to generate better responses for the new users and dyads.



(a) BLEU (b) Embedding Avg (c) ROUGE-L F

Figure 3: Response quality test on four cases of new users and dyads. The range of the vertical error bar is one standard error of the metric values among the experimental trials. VHUCM-PUE outperforms the other models in cases involving new user with a known user whose conversations are in the training data (*Known Partner* case).



(a) VHUCM-PUE with full data (b) VHUCM with removed data (c) VHUCM-PUE with removed data

Figure 4: Plots of two-dimensions projection of trained user vectors by each model and data by t-SNE. Four types of users: Known Users (●) who are in the *Known Dyad* and *Known Users* cases, New Users (▶) and Known Partners (◀) who are in the *Known Partner* case, and New Users (◆) who are in the *New Users* case. (a) The conversation partners (▶ and ◀) are closed in the embedding space. (b) Many new users (▶ and ◆) are not well distributed. (c) The conversation partners (▶ and ◀) are well paired as similar as (a).

Acknowledgments

We would like to thank the anonymous reviewers for helpful questions and comments. This work was supported by IITP grant funded by the Korea government (MSIT) (No.2017-0-01779, XAI).

References

JinYeong Bak, Chin-Yew Lin, and Alice Oh. 2014. Self-disclosure topic model for classifying and analyzing twitter conversations. In *Proceedings of the EMNLP*.

Parminder Bhatia, Marsal Gavaldà, and Arash Einolghozati. 2017. soc2seq: Social embedding meets conversation model. *arXiv preprint*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the CoNLL*.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the WMT*.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the CMCL*.

Jiachen Du, Wenjie Li, Yulan He, Ruifeng Xu, Lidong Bing, and Xuan Wang. 2018. Variational autoregressive decoder for neural response generation. In *Proceedings of the EMNLP*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the NIPS*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the SIGKDD*.

Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2019. DialogWAE: Multimodal response generation with conditional wasserstein auto-encoder. In *Proceedings of the ICLR*.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Proceedings of the NIPS*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting ob-

jective function for neural conversation models. In *Proceedings of the NAACL*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proceedings of the ACL*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. In *Proceedings of the EMNLP*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the IJCNLP*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Per Linell, Antony Manstead, et al. 1991. *Contexts of Accommodation: Developments in Applied Sociolinguistics*. Cambridge University Press.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the EMNLP*.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the SIGDIAL*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*.

Pierre-Emmanuel Mazare, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the EMNLP*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the NIPS*.

O. O. Olabiyi, A. Khazane, and E. T. Mueller. 2018. A persona-based multi-turn conversation model in an adversarial learning framework. In *Proceedings of the ICMLA*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*.

Yookoon Park, Jaemin Cho, and Gunhee Kim. 2018. A hierarchical latent structure for variational conversation modeling. In *Proceedings of the NAACL*.

Adwait Ratnaparkhi. 2002. Trainable approaches to surface natural language generation and their application to conversational dialog systems. *Computer Speech & Language*.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the EMNLP*.

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

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the CIKM*.

Yujie Xing and Raquel Fernández. 2018. Automatic evaluation of neural personality-based chatbots. In *Proceedings of the INLG*.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie SUN, Xiaolong Wang, Zhuoran Wang, and Chao Qi. 2017. Neural response generation via gan with an approximate embedding layer. In *Proceedings of the EMNLP*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the ACL*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the ACL*.