

Learning from My Friends: Few-Shot Personalized Conversation Systems via Social Networks

Zhiliang Tian,^{1,4*} Wei Bi^{2,†}, Zihan Zhang,¹ Dongkyu Lee,¹ Yiping Song,³ Nevin L. Zhang^{1,4}

¹Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology, Hong Kong SAR, China

²Tencent AI Lab, Shenzhen, China

³National University of Defense Technology, Changsha, China

⁴HKUST Xiao-i Robot Joint Lab, Hong Kong SAR, China

{ztianac,dleear,lzhang}@cse.ust.hk victoriabi@tencent.com zzhangeo@connect.ust.hk songyiping@pku.edu.cn

Abstract

Personalized conversation models (PCMs) generate responses according to speaker preferences. Existing personalized conversation tasks typically require models to extract speaker preferences from user descriptions or their conversation histories, which are scarce for newcomers and inactive users. In this paper, we propose a few-shot personalized conversation task with an auxiliary social network. The task requires models to generate personalized responses for a speaker given a few conversations from the speaker and a social network. Existing methods are mainly designed to incorporate descriptions or conversation histories. Those methods can hardly model speakers with so few conversations or connections between speakers. To better cater for newcomers with few resources, we propose a personalized conversation model (PCM) that learns to adapt to new speakers as well as enabling new speakers to learn from resource-rich speakers. Particularly, based on a meta-learning based PCM, we propose a task aggregator (TA) to collect other speakers' information from the social network. The TA provides prior knowledge of the new speaker in its meta-learning. Experimental results show our methods outperform all baselines in appropriateness, diversity, and consistency with speakers.

Introduction

Recently, there has been a boom in research on neural conversation models (Shang, Lu, and Li 2015) due to the accessibility of vast conversational data on social media (e.g. Twitter). To generate appropriate and lively responses, researchers have proposed personalized conversation tasks that require models to customize responses for specific speakers, since different speakers tend to have different styles or preferences for their responses. There are two subtypes of such tasks. Description-conditioned tasks (Yang et al. 2017; Zhang et al. 2018b) require models to customize responses according to explicit speaker descriptions. These descriptions may come from human annotations (Zhang

et al. 2018b) or user profiles in social media (Mazare et al. 2018). Speaker descriptions are not always available due to the cost of annotation and privacy concerns in social media. Conversation-conditioned tasks (Li et al. 2016b; Kottur, Wang, and Carvalho 2017) require models to generate personalized responses by exploiting speakers' preferences from their conversation histories. In reality, conversation histories may provide very few utterances of a particular speaker, which makes it hard to capture speaker preferences, especially for the newcomers or inactive users.

To better cater for newcomers, we propose a few-shot personalized conversation task with an auxiliary social network. The task has three characteristics: 1. During training, models cannot access information about the speakers in the testing set (i.e. newcomers); 2. During testing, there are only a few samples available for each speaker, which are collectively referred to as the support set; 3. There is a social network among all the speakers. Given the input query, our task requires a conversation model to generate a response for a new speaker with the help of the social network and a few (i.e. 10) past conversation samples from the speaker.

It is difficult to characterize the preferences of a new speaker from only a few conversation samples. Social networks can help here. In a social network, neighbors are users who follow each other, and they usually share interests and have similar chatting preferences. As our observation on our dataset, on average, the response similarity between two neighbors (0.47) is higher than that between two random speakers (0.38).¹ Consequently, we can utilize conversation histories of neighbors to help to determine preferences of a newcomer. In this way, we can handle newcomers even when there are no descriptions or a few conversations available.

Existing conversation-conditioned PCMs can be applied to our proposed task. Li et al. 2016b employ speaker embedding to capture speaker preferences. Based on this, Bak and Oh 2019 pre-built a conversation graph to describe speaker relations and learn node2vec embeddings (Grover and Leskovec 2016) over the graph. The node2vec embed-

*This work was partially done when Zhiliang Tian was an intern at Tencent AI Lab

[†]Corresponding author

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹The gap between 0.47 and 0.38 is quite large in this evaluation metric, as a contrast, the gap between similarity of two responses from one speaker (0.50) and 0.47 is only 0.03.

dings serve as the initial speaker embeddings in conversation models. To adopt such methods to our setting, we first train the models on the large scale training set; and then fine-tune them on a few conversations from the target speaker. However, training speaker embeddings or building a reliable conversation graph requires many samples from the target speaker, which is not always available in our setting. Model-agnostic meta-learning (MAML) (Finn, Abbeel, and Levine 2017) based PCMs (Madotto et al. 2019) is capable of fast adapting to new speakers by learning the adaptation ability across speakers. However, it aims to be a good initial model that is satisfactory for all speakers’ adaption, instead of caring about the speakers’ characteristics and relations. Due to the weakness in modeling speaker relations and distinguishing characteristics of speakers, this paradigm is not effective at making use of other resource-rich speakers.

In this paper, we propose a PCM that learns to adapt to new speakers as well as enables new speakers to leverage information from resource-rich speakers via a social network. Specifically, we first construct a MAML based PCM, where each speaker functions as a task. Then, we propose a task aggregator (TA) to collect resource-rich speakers’ information according to speaker relations in the social network. The task aggregator explicitly represents each task, and the task representation serves as a task prior for the new speaker in its meta-learning.

In this way, our model utilizes the information of resource-rich speakers to augment low-resource speakers, and hence addresses the data deficiency issue on low-resource speakers. The experimental results show that our methods outperforms all alternative methods in terms of appropriateness, informativeness, and consistency.

Our contributions are summarized as follows,

- We propose a few-shot personalized conversation task with a social network that better caters for new speakers.
- We design some variants of existing PCMs to our proposed task as strong baselines for our task.
- We propose a novel method that learns to learn from resource-rich speakers to customize responses for new speakers, and our model surpasses all the baselines.

Related Work

Conversation Models

Large scale conversation corpora from social media lead to a great success of retrieval-based methods (Yan, Song, and Wu 2016; Zhou et al. 2016) and generation-based methods (Sutskever, Vinyals, and Le 2014; Shang, Lu, and Li 2015; Zhao, Zhao, and Eskenazi 2017). To enhance the conversation models, Serban et al. 2016; Tian et al. 2017 proposed context-aware conversations that consider the previous utterances in current dialogue session. To ensure the topic consistency between the input and output, researchers proposed topic-aware models to capture and emphasize input’s topic (Xing et al. 2017; Peng et al. 2019). Some papers utilized the background knowledge (Moghe et al. 2018; Qin et al. 2019; Ren et al. 2020) and commonsense knowledge (Zhou et al. 2018b; Young et al. 2018; Tian et al. 2019) in conversations.

Researchers proposed to capture speakers’ temporal status to generate appropriate responses. Emotion-aware dialogue models detect speaker’s feelings and make appropriate emotional responses (Zhou et al. 2018a; Fung et al. 2018; Saha et al. 2020). Multi-party dialog (Meng, Mou, and Jin 2018; Zhang et al. 2018a; Hu et al. 2019) distinguished different roles of different speakers within a dialogue session.

Personalized Conversation Models

Personalized conversation models capture speaker’s characteristic, such as language behaviors, styles, and hobbies. (Li et al. 2016b; Zhang et al. 2018b; Olabiyi, Khazane, and Mueller 2018; Song et al. 2020; Zheng et al. 2020). Some researchers focused on the conversational agent being aware of the human’s personality and adjusting the dialogue accordingly (Lucas et al. 2009; Joshi, Mi, and Faltings 2017; Luo et al. 2019). Others assigned the personality to the conversational agent and encouraged the agent to generate personalized responses. Our task falls into the latter category, which consists of two types: description-conditioned and conversation-conditioned.

Description-conditioned personalized conversation tasks capture the speaker preference by explicit speaker description. Zhang et al. 2018b described the speaker with a few profile sentences and built Persona-Chat dataset by crowdsourcing. Mazare et al. 2018 summarized speaker comments from Reddit into several sentences as speaker descriptions. Qian et al. 2018; Zheng et al. 2019 defined some profile key-value pairs, such as gender, age, and location. As for the methods on the above tasks, Yang et al. 2017 fine-tuned a speaker-specific model based on a pre-trained speaker-independent model. Zhang et al. 2018b encoded persona description into a memory network and employed a seq2seq model to generate responses with the memory, and Chu, Vijayaraghavan, and Roy 2018 refined the memory network with attention mechanism (Bahdanau, Cho, and Bengio 2015). Yavuz et al. 2019 explored the use of copy mechanism in personalized dialogue models. Zheng et al. 2020 imported a pre-trained GPT to transformer-based personalized dialog models. Different from this type of task, our task does not require the external description of speakers.

Conversation-conditioned personalized conversation tasks describe speakers with their conversation histories. Li et al. 2016b, the first to propose neural personalized conversation models, implicitly modeled speakers’ personas using speaker embeddings. Kottur, Wang, and Carvalho 2017 modeled the speaker personality together with conversational contexts by a hierarchical recurrent encoder-decoder model (Serban et al. 2016) with speaker embeddings. Bak and Oh 2019 learned node2vec (Grover and Leskovec 2016) embeddings from a conversation graph to initiate the speaker embeddings. The above three papers cannot handle new speakers, and they require many dialogues for each speaker. Our task is designed to handle new speakers with very few resources. Madotto et al. 2019 learned to adapt to new speakers with few shots, but their framework is not effective at incorporating and utilizing speaker relations. Our model caters for new speakers and encourages resource-rich speakers to help low-resource speakers in a social network.

Task Definition

Our task aims to enable conversation systems to act as a given speaker and make personalized responses to the input query according to the speaker preference. Our task requires the systems to follow a speaker preference or style, even if the speaker is a newcomer with little information.

Our setting consists of a training set, a testing set, and a validation set. Each of those sets consists of speakers $\mathbb{S} = \{s_1, s_2, \dots, s_n\}$, the conversation samples $D_s = \{\langle q, r_s \rangle_1, \langle q, r_s \rangle_2, \dots, \langle q, r_s \rangle_m\}$ involving speaker s , and a graph \mathcal{G} that represents the social network. The set is denoted as $\{\mathbb{S}, \bigcup_{s \in \mathbb{S}} D_s, \mathcal{G}\}$. A conversation sample $\langle q, r_s \rangle$ is a pair of input queries and the corresponding responses made by speaker s . In the undirected graph \mathcal{G} , speakers act as the nodes, relations between speakers act as the edges, and two nodes bridge an edge once two speakers follow one another.

The speakers in the testing or validation set are newcomers, and they do not overlap with speakers in the training set. The graph in the training set is composed of only the training speakers, and the graph in testing or validation contains all speakers. For each newcomer s , we can access only K conversation samples before inference (prediction), which make up the support set D_s^{sup} . We are required to make inference on other K conversation samples from the speaker, which make up the query set D_s^{qu} . K is the shot number for few-shot learning. In summary, our task is to generate a response given a query q , a current speaker s , the speaker’s support set D_s^{sup} , and a social network \mathcal{G} .

Methodology

Architecture

Our model consists of two sub-modules as shown in Fig. 1:

- *Personalized Conversation Model (PCM)* makes personalized responses according to speaker preferences. A transformer-based speaker-independent conversation model (CM) acts as the base model of PCM, denoted as f_ϕ^{CM} with its parameters ϕ . PCM employs MAML to learn to adapt CM to new tasks (i.e. speakers) with few examples so as to make personalized responses.
- *Task Aggregator (TA)* learns to study from related tasks (i.e. speakers) to assist the target task. Based on PCM, The TA explicitly represents tasks with embeddings, and then learns to aggregate embeddings from other tasks to refine the representation of the target task. The refined representation serves as the prior knowledge of the target task in PCM. We denote the operation of obtaining task embeddings as f_v^{Emb} with its parameters v , and operation of aggregating related task embeddings as f_φ^{AG} with parameters φ .

The TA provides the prior knowledge of the target task for PCM, while PCM provides the feedback for the TA’s back-propagation. The TA regards each task as a training sample, and PCM’s performance on an entire task serves as the feedback for a TA’s training step. We introduce the two sub-modules and our training and testing methods in the following subsections.

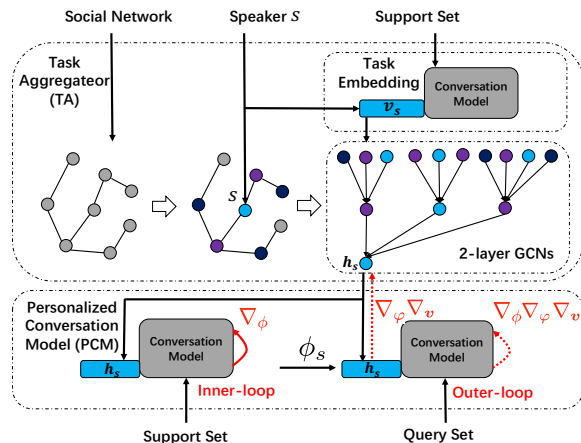


Figure 1: The structure of our model. Solid and dashed red arrows indicate the backward pass of PCM’s inner-loop and outer-loop, respectively. Blue, purple, and mazarine blue blocks represent variables of speaker s , its 1-hop neighbors, and its 2-hop neighbors, respectively. The whole figure shows the training mode; in the testing mode, operations shown by dashed red arrows do not work. In our full model, all components in the figure works; in our variant of *Ours-SelfEmb*, the task embedding block does not work. Eq. 2 to 5 show the optimization of ϕ , φ , and v in the inner- and outer-loops.

Personalized Conversation Model (PCM)

PCM learns to generate personalized responses for new speakers via MAML (Finn, Abbeel, and Levine 2017), where it treats new speakers as new tasks. First, PCM constructs a speaker-independent conversation model (CM). The CM is a conventional transformer model (Vaswani et al. 2017) with the ability to generate response r to the input query q . We formalize the CM as $r = f_\phi^{CM}(q)$. Then, we apply MAML to the CM to build a personalized model. MAML can learn good initial parameters for CM that achieves rapid adaptation to new speakers with a few examples. For each training step, we sample a batch of speakers with their samples $\{S = \{s_1, \dots, s_l\}, \bigcup_{s \in S} D_s\}$, and then build a support set D_s^{sup} and query set D_s^{qu} with K distinct examples sampled from D_s , respectively.

The training of MAML consists of an inner-loop phase and an outer-loop phase. In the inner-loop, MAML trains the conversation model f_ϕ^{CM} on D_s^{sup} and adapts f_ϕ^{CM} to the speaker s by updating its parameters from ϕ to ϕ_s . In the outer-loop, MAML updates the parameters ϕ according to the performance of the model trained by the inner-loop phase, which is evaluated on the query set. Particularly, in the inner-loop, the parameters ϕ is optimized by the gradient of $f_{\phi_s}^{CM}$ with respect to ϕ across tasks in S .

Task Aggregator (TA)

While PCM customizes responses for the given single task, TA models task representations and relations among tasks with the purpose of providing the prior knowledge of a task

for PCM. TA obtains conversation-conditioned task embeddings by f_v^{Emb} , refines task embeddings with the social network and a task embedding aggregator f_φ^{AG} , and then feeds the refined task embeddings to PCM. TA is optimized according to PCM’s feedback; meanwhile, the task embeddings are also constrained by the graph structure of the social network.

Conversation-conditioned Task Embeddings We employ a task embedding layer, assign each task with an embedding, and learn the task embedding with conversations under this task. As traditional embedding layers, task embeddings start from randomly initialized parameters. During training, we fetch the task embedding v_s of the current task s , and combine the embedding with the conversation model (CM) by appending the embedding in front of the sequence of CM’s input word embeddings.² Afterwards, we train the combined model on the conversations from the support set D_s^{sup} of the current task s . After the training, the task embedding v_s serves as the output of this component. We formulate this component as $v_s = f_v^{Emb}(s, D_s^{sup})$, where the inputs are current speaker s and its support set, and v denotes the task embedding parameters.

Task embedding parameters v and CM’s parameters ϕ are optimized iteratively. In the training of this component, we fix ϕ and optimize the embedding v ; in the training of PCM and we optimize CM’s parameters ϕ . (Sec.).

Task Embedding Aggregator via GCN As the scale of D_s^{sup} for new speakers is still too small for the task embedding learning, we enhance the task embedding by related resource-rich tasks.

Social networks naturally reveal the relationship between speakers. We construct an undirected graph \mathcal{G} , where speakers act as nodes, and two nodes bridge an edge once two speakers follow one another. Since we observed that neighbors (two speakers follow each other) have similar preferences in responding, we utilize task embeddings of neighboring speakers seen in the training set.

We employ 2-layer GCNs (Kipf and Welling 2017) to refine task embeddings with neighbors’ embeddings. The original GCN (Kipf and Welling 2017) conducts the propagation on the whole graph. It is time-consuming to re-train with the whole graph for the newcomers, so we follow the neighbor aggregation variant of GCN (Hou et al. 2019). We formulate this aggregator as $h_s = f_\varphi^{AG}(v_s, \mathcal{G})$.

We also propose another model variant that does not acquire the conversation-conditioned task embedding v_s of the target task and relies only on the neighbors. That variant gets the final task representation h_s by feeding all the neighbor’s embedding into the GCN without using v_s . We discuss it in the experiment section with the notation of *Ours–SelfEmb*.

Graph Structure Constraint via Negative Sampling f_v^{Emb} represents a task according to conversations. We argue that task representations should also be constrained by the graph \mathcal{G} , which reveals task relations. As neighbors share

²We did try some different combinations of extra embeddings with the transformer model and picked the best one, and similar usage can be found in (Keskar et al. 2019).

Algorithm 1: Training Algorithm

Input: $\{\mathbb{S}, \bigcup_{s \in \mathbb{S}} D_s\}$: training set
 α, β : learning rates
 λ : weight for negative sampling loss
 \mathcal{G} : social network on training speakers

Output: $\{\phi, \varphi, v\}$: trained model parameters
Randomly initialize ϕ, φ, v .

while not done do

- Sample a batch of speakers with their samples
 $\{S, \bigcup_{s \in S} D_s\} \sim \{\mathbb{S}, \bigcup_{s \in \mathbb{S}} D_s\}$
- for all s in S do**
 - // Inner-loop (train the base model)
 - Build D_s^{sup} and D_s^{qu} from D_s as Sec.
 - Get task embeddings v_s by f_v^{Emb}
 - Get aggregated task embeddings h_s via GCN
 - Evaluate $\nabla_\phi \mathcal{L}_{D_s^{sup}}(f_\phi^{CM}(h_s, q))$ with respect to K examples.
 - Compute adapted parameters ϕ_s with gradient descent as Eq. 2
- // Outer-loop (meta-optimization across tasks)
- Evaluate $\nabla_{\{\phi, \varphi, v\}} \sum_{s \sim S} \mathcal{L}_{D_s^{qu}}(f_{\phi_s}^{CM}(f_\varphi^{AG}(f_v^{Emb}(\cdot))))$ with respect to K examples.
- Evaluate $\nabla_{\{\varphi, v\}} \lambda \sum_{s \sim S} \mathcal{L}_s^{ns}(f_\varphi^{AG}(f_v^{Emb}(\cdot)))$ with respect to \mathcal{G} .
- Update $\{\phi, \varphi, v\}$ using Eq. 3, 4, and 5.

similar preferences, they ought to have similar task embeddings. Inspired by node2vec (Grover and Leskovec 2016), we employ a negative sampling loss (Mikolov et al. 2013) that forces task embeddings from two neighbors to be more similar than those from two randomly selected speakers as,

$$\mathcal{L}_s^{ns} = -\left(\sum_{i \in \mathcal{N}_s} \log \sigma(v_i^T h_s) + \sum_{\substack{j \sim \mathbb{S} \\ k=1}}^{K_{n,s}} \log \sigma(-v_j^T h_s)\right) \quad (1)$$

where h_s comes from the aggregator, $K_{n,s}$ denotes the number of negative samples, σ denotes the sigmoid function, and \mathbb{S} denotes the set of all speakers. Notice that the above two algorithms involving the graph are friendly to newcomers since their training only operates on the newcomers and their neighbors instead of conducting the propagation over the entire graph.

Training and Testing

Model Training. Our whole model is the combination of the PCM and TA. We extend the PCM’s training paradigm mentioned in Sec. to the whole model. We train the PCM via MAML as in Sec. , and train the TA in MAML’s outer-loop. Algorithm 1 and Fig. 1 show the training procedure.

During the training, the TA first obtains the task embedding h_s , and then feeds it to the PCM. The PCM training consists of inner-loop and outer-loop training. In the inner-loop, the PCM uses h_s in its feedforward but does not update the TA’s parameters. In the outer-loop, the PCM uses h_s again and updates the parameters of both the PCM and

TA according to the performance of the whole model on the query set. The optimization of the inner-loop and outer-loop is shown in Eq. 2 to 5. In both inner- and outer-loop training, PCM uses \mathbf{h}_s by appending it in front of the sequence of CM’s input word embeddings, where the formula of CM defined in Sec. turns to $r = f_{\phi}^{CM}(\mathbf{h}_s, q)$.

TA is to model task priors and relations among tasks, so we feed each task as a training sample to the TA. Then, we train and optimize the TA on a set of tasks. Since inner-loop trains samples within a task and outer-loop optimizes across tasks, we fix the TA’s parameters in the inner-loop and optimize them in the outer-loop. In the outer-loop, the TA receives feedback about whether the current TA with the PCM performs well on each task and is optimized accordingly.

$$\phi_s = \phi - \alpha \nabla_{\phi} \mathcal{L}_{(q,r) \sim D_s^{sup}}(f_{\phi}^{CM}(\mathbf{h}_s, q)). \quad (2)$$

$$\phi \leftarrow \phi - \beta \nabla_{\phi} \sum_{s \sim S} \mathcal{L}_{D_s^{qu}}(f_{\phi_s}^{CM}(\mathbf{h}_s, q)), \quad (3)$$

$$\varphi \leftarrow \varphi - \beta \nabla_{\varphi} \sum_{s \sim S} (\lambda \mathcal{L}_s^{ns} \quad (4)$$

$$+ \mathcal{L}_{D_s^{qu}}(f_{\phi_s}^{CM}(f_{\varphi}^{AG}(\mathbf{v}_s, \mathcal{G}), q))),$$

$$\mathbf{v} \leftarrow \mathbf{v} - \beta \nabla_{\mathbf{v}} \sum_{s \sim S} (\lambda \mathcal{L}_s^{ns} \quad (5)$$

$$+ \mathcal{L}_{D_s^{qu}}(f_{\phi_s}^{CM}(f_{\varphi}^{AG}(f_{\mathbf{v}}^{Emb}(s, D_s^{sup}), \mathcal{G}), q))).$$

Model Testing. Model testing adopts the training procedure except for the back-propagation in the outer-loop. The TA obtains the task representation \mathbf{h}_s for the PCM. The PCM incorporates \mathbf{h}_s , updates its ϕ to ϕ_s in the inner-loop, and generates the final responses with the feedforward of the outer-loop. The only difference between training and testing is that testing does not need not to optimize parameters in the outer-loop (No dashed red arrows in Fig 1 for testing).

Experimental Setting

Dataset

We collect the dataset from Weibo, an online chatting forum with social networks. We use 28.9K speakers with 2.02M samples for training, 1K speakers with 20K samples for testing, and 0.5K speakers with 10K samples for validation. We release the code of dataset construction³.

Comparing Methods

- **Base Models.** We use two conventional conversation models, *Seq2Seq* and *Transformer*. We use the transformer as our base conversation model for all the following methods as it performs better than the seq2seq.
- **Fine-tune.** We fine-tune the transformer on the support set of the target speaker to obtain personalized models, noted as *Transformer+F*. We increase the hidden dimension of the transformer to keep the same parameter scale with the following comparing models, noted

as *Transformer⁺+F*. We employ MAML (Finn, Abbeel, and Levine 2017) to train the transformer and note it as *PAML* (Madotto et al. 2019).

- **Fine-tune+Social Network.** Li et al. 2016b encode speaker preferences with speaker embeddings. Bak and Oh 2019 pre-train node2vec embeddings over a speaker graph as the initial speaker embeddings. Since new speakers are unseen during training, their methods cannot obtain new speakers’ embeddings. We adapt their original methods to our task by aggregating its neighbors’ embeddings and then fine-tune the embeddings on the support set. We denote them as *Speaker+F+SN* and *VHUCM+F+SN*, respectively.

- **Ours.** *Ours* denotes our full model, and *Ours-SelfEmb* is our model without using the conversation-conditioned embedding of the target speaker.

Implementation Details

Seq2Seq follows Song et al. 2018 where the embedding and hidden dimensions are 620 and 1000. For the transformer-based model, we implement it as the original one (Vaswani et al. 2017), where the model dimension is 512, the stacked layer number is 6, and the head number is 8. We increase the model dimension to 640 on *Transformer⁺+F* for fair comparison on the parameter scale. Following (Madotto et al. 2019), in *PAML* and ours, we used SGD for the inner-loop and Adam for the outer-loop with learning rate $\alpha = 0.01$ and $\beta = 0.0003$, respectively. For all methods, the batch size in training is 128. The vocabulary contains top 50k frequent tokens, and the maximum length of input queries and responses is 80. For training, we set λ as 1 for the negative sampling. For inference, we apply a top- k sampling decoding (Edunov et al. 2018) with $k=5$. The sizes of the support set and query set for testing users are 10 (10-shot).

Evaluation Metrics

We evaluate all methods with both automatic and human evaluations. For automatic evaluations, we evaluate the responses in three aspects:

1. **Appropriateness.** We evaluate the overall quality by measuring the matching between the ground-truth and generated responses on three metrics: *BLEU* (Papineni et al. 2002), *NIST* (Doddington 2002), and *CIDEr* (Vedantam, Lawrence Zitnick, and Parikh 2015).
2. **Diversity.** *Dist-n* evaluates the proportion of n-grams of the generated responses (Li et al. 2016a; Song et al. 2017).
3. **Consistency.** We measure the consistency between the generated responses and the speakers’ conversation histories. *TokSim* measures the consistency by token-level similarity with TF-IDF (Joachims 1996) weight and a stop-word filter. F1 grounding score *Grd-F1* (Qin et al. 2019; Tian et al. 2020) is the harmonic mean of precision and recall, where precision and recall measures the ratio of retrieving a token from conversation histories as responses.

For human evaluations, we hire five annotators from a commercial company to evaluate 300 samples randomly selected from 60 speakers. The annotators evaluate a 5-point

³github.com/tianzhiliang/FewShotPersonaConvData

| | Appropriateness | | | | | | Diversity | | Consistency | | Param |
|-----------------------------|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-------|
| | Bleu1 | Bleu2 | Bleu3 | Bleu4 | NIST | CIDEr | Dist1 | Dist2 | Grd-F1 | TokSim | |
| Seq2Seq | 7.325 | 3.912 | 3.129 | 2.736 | 0.329 | 0.149 | 0.133 | 0.370 | 0.012 | 0.041 | 62M |
| Transformer | 10.951 | 5.852 | 4.407 | 3.803 | 0.803 | 0.205 | 0.155 | 0.454 | 0.016 | 0.032 | 65M |
| Transformer+F | 11.109 | 5.902 | 4.466 | 3.459 | 0.790 | 0.182 | 0.157 | 0.485 | 0.020 | 0.052 | 65M |
| Transformer ⁺ +F | 12.219 | 5.992 | 4.486 | 3.643 | 0.880 | 0.189 | 0.160 | 0.488 | 0.020 | 0.048 | 80M |
| PAML | 11.200 | 6.192 | 4.691 | 4.012 | 0.884 | 0.201 | 0.174 | 0.495 | 0.021 | 0.054 | 80M |
| Speaker+F+SN | 11.670 | 6.182 | 4.615 | 3.922 | 0.927 | 0.176 | 0.133 | 0.427 | 0.018 | 0.035 | 80M |
| VHUCM+F+SN | 10.981 | 5.461 | 3.976 | 3.357 | 0.892 | 0.141 | 0.142 | 0.460 | 0.018 | 0.043 | 80M |
| Ours–SelfEmb | 11.815 | 6.277 | 4.676 | 4.010 | 0.971 | 0.196 | 0.183 | 0.586 | 0.023 | 0.061 | 80M |
| Ours | 11.638 | 6.315 | 4.803 | 4.145 | 0.940 | 0.238 | 0.169 | 0.530 | 0.024 | 0.062 | 80M |

Table 1: The overall performance on automatic evaluations.

| | Appropriateness | | | | | | Diversity | | Consistency | |
|---------------------|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Bleu1 | Bleu2 | Bleu3 | Bleu4 | NIST | CIDEr | Dist1 | Dist2 | Grd-F1 | TokSim |
| Ours | 11.638 | 6.315 | 4.803 | 4.145 | 0.940 | 0.238 | 0.169 | 0.530 | 0.024 | 0.062 |
| –NS Loss | 11.002 | 5.784 | 4.343 | 3.730 | 0.954 | 0.188 | 0.167 | 0.550 | 0.022 | 0.053 |
| –GCN | 11.176 | 6.959 | 4.425 | 3.782 | 0.870 | 0.194 | 0.158 | 0.526 | 0.022 | 0.053 |
| –NS Loss –GCN | 10.877 | 5.338 | 4.002 | 3.439 | 0.615 | 0.194 | 0.167 | 0.510 | 0.020 | 0.052 |
| –NS Loss –GCN + RAG | 10.989 | 5.771 | 4.202 | 3.552 | 0.779 | 0.192 | 0.175 | 0.491 | 0.020 | 0.048 |
| TA as MAML’s base | 10.642 | 5.359 | 2.926 | 3.342 | 0.790 | 0.169 | 0.165 | 0.548 | 0.021 | 0.055 |

Table 2: The performance of the ablation study. Row 2 to row 4 indicates the full model without the negative sampling, without the GCN layer, or without both of them. + RAG indicates applying retrieval-augmented generation methods. TA as MAML’s base optimizes TA by treating it as a part of MAML’s base model.

scale on quality (*H-Appr*), diversity (*H-Div*), and consistency with speakers’ previous conversations (*H-Cons*).

Experimental Results and Analysis

Overall Performance

The results of all competing methods on automatic metrics are shown in Table 1. Since *Transformer* outperforms *Seq2Seq*, we use *Transformer* as the base generation model for all baselines. **Fine-tune** methods achieve the better performance than **Base models** methods, indicating that building personalized conversation models for each speaker helps to improve the overall performance. *PAML* gets higher performance than other baselines, which demonstrates that meta-learning is more suitable to handle new speakers in our scenarios. In rows 4 to 9 of Table 1, we compare all methods in the same scale of parameters (80M). Social network describes the relations among speakers, so the baseline methods of **Fine-tune+Social Network** are supposed to be better than **Fine-tune**. However, it is not the case in our experiments, which demonstrates that few conversations are inapplicable for training a speaker embedding, so the neighbors’ information cannot contribute to the target speaker with the usages in **Fine-tune+Social Network**.

Ours methods outperform all baselines on most metrics, especially on the consistency. We conclude that resource-rich neighbors are qualified to provide task priors for low-resource speakers. For our two variants, our full model obtains higher scores on the overall quality and the consistency to the speakers. *Ours–SelfEmb*, which skips the training for the target speaker’s embeddings, generates more diverse results. This is because absorbing the information from neighbors promotes diversity, and utilizing the embeddings

| | H-Appr | H-Div | H-Cons |
|-----------------------------|-------------|-------------|-------------|
| Seq2Seq | 2.77 | 1.84 | 2.27 |
| Transformer | 2.98 | 2.87 | 2.67 |
| Transformer+F | 3.09 | 2.78 | 2.45 |
| Transformer ⁺ +F | 3.15 | 2.72 | 2.88 |
| PAML | 3.05 | 2.68 | 3.02 |
| Speaker+F+SN | 2.91 | 2.66 | 3.02 |
| VHUCM+F+SN | 2.99 | 2.56 | 2.84 |
| Ours–SelfEmb | 3.09 | 2.84 | 2.97 |
| Ours | 3.17 | 2.79 | 3.09 |

Table 3: The overall performance on human evaluations.

trained on its conversations helps the model to concentrate more on the target speaker’s unique characteristics.

The human evaluation results in Table 3 are almost consistent with the automatic metrics. The only exception is that *Transformer* has the highest diversity score, but our methods obtain the second highest score in diversity.

Ablation Study

We demonstrate the performance of the proposed components with an ablation study shown in Table 2. *–NS Loss* shows that removing negative sampling loss from our model causes a performance drop, especially on the appropriateness and consistency. Hence, it is necessary to restrict the task embeddings with the graph structure. *–GCN* aggregates the neighbor embeddings by simply averaging it instead of the GCN operation. The performance of replacing the GCN layer also decreases, which verifies the GCN’s effect. The model lacking both negative sampling and GCN (*–NS Loss –GCN*) becomes much worse. *–NS Loss –GCN*

+ *RAG* means the variant that lacks negative sampling and GCN and follows the retrieval-augmented generation idea (Song et al. 2018) to retrieve similar samples from neighbors’ conversation samples. *TA as MAML’s base* treats TA as a part of MAML’s base model and optimizes TA together with CM instead of optimizing TA across tasks in the outer-loop. Its bad performance suggests that as a cross-task component to capture task embedding and task relation, TA should be optimized across tasks.

Quality of New Speakers’ Task Embeddings

In this section, we examine the quality of new speakers’ refined task embedding h_s by verifying their ability to distinguish similar speakers from dissimilar ones. As social network neighbors tend to share similar preferences, we regard neighbors to be similar than randomly paired speakers. For each new speaker s , we calculate two task embedding similarity scores: the similarity of s and one of its neighbor s_n , denoted as $sim(s, s_n)$; the similarity of s and a randomly picked speaker s_r , denoted as $sim(s, s_r)$. We measure a ratio of $sim(s, s_n)$ being higher than $sim(s, s_r)$ plus a margin c , which is defined as $\gamma-c = \frac{\#\{sim(s, s_n) > sim(s, s_r) + c\}}{|s|}$.

| | $\gamma-0$ | $\gamma-0.1$ | $\gamma-0.2$ |
|-------------------------|------------|--------------|--------------|
| Ours (TrainSpeakers) | 0.99 | 0.99 | 0.89 |
| Ours (RichTestSpeakers) | 0.99 | 0.96 | 0.83 |
| Ours | 0.99 | 0.95 | 0.82 |
| VHUCM+F+SN | 0.96 | 0.83 | 0.51 |
| Speaker+F+SN | 0.91 | 0.20 | 0.03 |

Table 4: The quality of new speakers’ task embeddings. $\gamma-c$ is the ratio of new speakers’ embeddings achieving higher similarity to neighbors rather than random speaker pairs with a margin c .

In Table 4, *Ours (TrainSpeakers)* measures $\gamma-c$ on a set of training speakers. *Ours (RichTestSpeakers)* measures $\gamma-c$ on new speakers never seen in training, but each speaker can access 60 samples in its support set. *Ours*, *VHUCM+F+SN*, and *Speaker+F+SN* follow the same setting as Sec. , where models measure $\gamma-c$ on new speakers never seen in training and access K (i.e. 10) samples in their support sets. We reconstruct the training and testing set and re-train the models, so that we can measure the above five methods on the same set of speakers, which consists of 1,000 speakers.

The results in Table 4 show that the quality of new speakers’ task embeddings from our model are much higher than that from the baselines. *Ours (TrainSpeakers)* provides the upper-bound of the task embedding quality since those embeddings are trained by the model, and each speaker has many samples in training. The gap between *Ours (RichTestSpeakers)* and *Ours* is quite small, which reveals that our methods help low-resource speakers to achieve a very close performance to the resource-rich speakers.

Case Study

Table 2 shows a case from the test set. In this sample, *Seq2Seq* and *Transformer+F* give general responses. *Transformer* and *Transformer⁺+F* capture a part of the informa-

| | |
|---|--|
| Historical responses of speaker s | 1.Your words can always heal my broken heart! 2.Thanks! Collect the pictures! 3.Found a nice fashion blogger, come on ladies! |
| Historical responses of a s ’s neighbor | 1.Let’s download the pictures. 2.The pictures are really funny . Download them. 3.Dare we cosplay the personages in Naruto ? |
| Query | My ex-boy friend used to tell me his mom he likes me, after breaking up , I know the meaning of that sentence . |
| Seq2Seq | I do think so, I do think so. |
| Transformer | I also want to break up . |
| Transformer+F | What did you say, who are you, who are you. |
| Transformer ⁺ +F | You’re no longer alone |
| PAML | I know that, and so as my boy friend! I am also an excellent and nice man. |
| Speaker+F+SN | Finally I break up , after breaking up, I know what the meaning of that sentence . |
| VHUCM+F+SN | First of all, you should have a girl friend . |
| Ours-SelfEmb | We love him no matter breaking up or not, right? This is the world for girls . All the girls like cartoon are here. |
| Ours | Your heart is broken . See some funny pictures! |

Figure 2: A case with generated responses of all models. We highlight semantic overlaps between generated responses and the query in red color, the responses and speaker’s conversation histories in blue, and the responses and neighbors’ conversation histories in cyan.

tion in query and make related responses. *Speaker+F+SN* almost repeats the query. *VHUCM+F+SN* outputs a humorous, relevant, and informative response, but it misunderstands the gender of the speaker s (s is probably a girl according to her third historical conversation). *PAML* acts as a girl at the beginning of its responses. As for our methods, *Ours-SelfEmb* makes appropriate responses in the view of a girl. Moreover, its response borrows information from the neighbor, where that information (“Naruto” is a “cartoon”) does not appear in the query or the speaker s ’s conversations. *Ours* can even study some relevant phases from other conversations of the speaker.

Conclusion

We propose a few-shot personalized conversation task with a social network, catering for low-resource speakers. In such a scenario, we propose a novel method which enables low-resource speakers to leverage information from the resource-rich speakers with the social network. Our method consists of a PCM and a TA. PCM equips conversational agents with personality via MAML; TA obtains the speaker representation from its few conversations and its resource-rich neighbors in the social network. The representation assists the training of the PCMs. In this way, we remedy the data deficiency issue on low-resource speakers so that our systems are friendly to newcomers or inactive users on social media. The experimental results show our model outperforms all baselines on the overall query, diversity, and consistency.

Acknowledgments

Research on this paper was supported by Hong Kong Research Grants Council under grants 16202118 and Ten-cent AI Lab Rhino-Bird Focused Research Program (No. GF202035).

References

- Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
- Bak, J.; and Oh, A. 2019. Variational hierarchical user-based conversation model. In *EMNLP-IJCNLP*, 1941–1950.
- Chu, E.; Vijayaraghavan, P.; and Roy, D. 2018. Learning personas from dialogue with attentive memory networks. In *EMNLP*, 2638–2646.
- Doddington, G. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *HLT*, 138–145.
- Edunov, S.; Ott, M.; Auli, M.; and Grangier, D. 2018. Understanding back-translation at scale. In *EMNLP*, 489–500.
- Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, 1126–1135.
- Fung, P.; Bertero, D.; Xu, P.; Park, J. H.; Wu, C.-S.; and Madotto, A. 2018. Empathetic dialog systems. In *LREC*.
- Grover, A.; and Leskovec, J. 2016. Node2vec: scalable feature learning for networks. In *SIGKDD*, 855–864.
- Hou, Y.; Zhang, J.; Cheng, J.; Ma, K.; Ma, R. T.; Chen, H.; and Yang, M.-C. 2019. Measuring and improving the use of graph information in graph neural networks. In *ICLR*.
- Hu, W.; Chan, Z.; Liu, B.; Zhao, D.; Ma, J.; and Yan, R. 2019. GSN: a graph-structured network for multi-party dialogues. In *AAAI*, 5010–5016.
- Joachims, T. 1996. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science.
- Joshi, C. K.; Mi, F.; and Faltings, B. 2017. Personalization in goal-oriented dialog. In *arXiv preprint arXiv:1706.07503*.
- Keskar, N. S.; McCann, B.; Varshney, L. R.; Xiong, C.; and Socher, R. 2019. Ctrl: A conditional transformer language model for controllable generation. In *arXiv preprint arXiv:1909.05858*.
- Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- Kottur, S.; Wang, X.; and Carvalho, V. 2017. Exploring personalized neural conversational models. In *IJCAI*, 3728–3734.
- Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016a. A diversity-promoting objective function for neural conversation models. In *NAACL*, 110–119.
- Li, J.; Galley, M.; Brockett, C.; Spithourakis, G.; Gao, J.; and Dolan, B. 2016b. A persona-based neural conversation model. In *ACL*, 994–1003.
- Lucas, J.; Fernández, F.; Salazar, J.; Ferreiros, J.; and San Segundo, R. 2009. Managing speaker identity and user profiles in a spoken dialogue system. In *Procesamiento del lenguaje natural*, 43, 77–84.
- Luo, L.; Huang, W.; Zeng, Q.; Nie, Z.; and Sun, X. 2019. Learning personalized end-to-end goal-oriented dialog. In *AAAI*, 6794–6801.
- Madotto, A.; Lin, Z.; Wu, C.-S.; and Fung, P. 2019. Personalizing dialogue agents via meta-learning. In *ACL*, 5454–5459.
- Mazare, P.-E.; Humeau, S.; Raison, M.; and Bordes, A. 2018. Training millions of personalized dialogue agents. In *EMNLP*, 2775–2779.
- Meng, Z.; Mou, L.; and Jin, Z. 2018. Towards neural speaker modeling in multi-party conversation: The task, dataset, and models. In *LREC*, 3142–3145.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 3111–3119.
- Moghe, N.; Arora, S.; Banerjee, S.; and Khapra, M. M. 2018. Towards exploiting background knowledge for building conversation systems. In *EMNLP*, 2322–2332.
- Olabiyyi, O. O.; Khazane, A.; and Mueller, E. T. 2018. A persona-based multi-turn conversation model in an adversarial learning framework. In *ICMLA*, 489–494. IEEE.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 311–318.
- Peng, Y.; Fang, Y.; Xie, Z.; and Zhou, G. 2019. Topic-enhanced emotional conversation generation with attention mechanism. In *KBS*, volume 163, 429–437. Elsevier.
- Qian, Q.; Huang, M.; Zhao, H.; Xu, J.; and Zhu, X. 2018. Assigning personality/identity to a chatting machine for coherent conversation generation. In *IJCAI-ECAI*, 4279–4285.
- Qin, L.; Galley, M.; Brockett, C.; Liu, X.; Gao, X.; Dolan, B.; Choi, Y.; and Gao, J. 2019. Conversing by reading: contentful neural conversation with on-demand machine reading. In *ACL*, 5427–5436.
- Ren, P.; Chen, Z.; Monz, C.; Ma, J.; and de Rijke, M. 2020. Thinking globally, acting locally: distantly supervised global-to-local knowledge selection for background based bonversation. In *AAAI*, 8697–8704.
- Saha, T.; Patra, A.; Saha, S.; and Bhattacharyya, P. 2020. Towards emotion-aided multi-modal dialogue act classification. In *ACL*, 4361–4372.
- Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, 3776–3783.
- Shang, L.; Lu, Z.; and Li, H. 2015. Neural responding machine for short-text conversation. In *ACL*, 1577–1586.

- Song, Y.; Li, C.-T.; Nie, J.-Y.; Zhang, M.; Zhao, D.; and Yan, R. 2018. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *IJCAI*, 4382–4388.
- Song, Y.; Liu, Z.; Bi, W.; Yan, R.; and Zhang, M. 2020. Learning to customize model structures for few-shot dialogue generation tasks. In *ACL*, 5832–5841.
- Song, Y.; Tian, Z.; Zhao, D.; Zhang, M.; and Yan, R. 2017. Diversifying neural conversation model with maximal marginal relevance. In *IJCNLP*, 169–174.
- Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*, 3104–3112.
- Tian, Z.; Bi, W.; Lee, D.; Xue, L.; Song, Y.; Liu, X.; and Zhang, N. L. 2020. Response-Anticipated Memory for On-Demand Knowledge Integration in Response Generation. In *ACL*, 650–659.
- Tian, Z.; Bi, W.; Li, X.; and Zhang, N. L. 2019. Learning to abstract for memory-augmented conversational response generation. In *ACL*, 3816–3825.
- Tian, Z.; Yan, R.; Mou, L.; Song, Y.; Feng, Y.; and Zhao, D. 2017. How to make context more useful? an empirical study on context-aware neural conversational models. In *ACL*, 231–236.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.
- Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: consensus-based image description evaluation. In *CVPR*, 4566–4575.
- Xing, C.; Wu, W.; Wu, Y.; Liu, J.; Huang, Y.; Zhou, M.; and Ma, W.-Y. 2017. Topic aware neural response generation. In *AAAI*, 3351–3357.
- Yan, R.; Song, Y.; and Wu, H. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *SIGIR*, 55–64.
- Yang, M.; Zhao, Z.; Zhao, W.; Chen, X.; Zhu, J.; Zhou, L.; and Cao, Z. 2017. Personalized response generation via domain adaptation. In *SIGIR*, 1021–1024.
- Yavuz, S.; Rastogi, A.; Chao, G.-L.; and Hakkani-Tur, D. 2019. DeepCopy: grounded response generation with hierarchical pointer networks. In *SIGDIAL*, 122–132.
- Young, T.; Cambria, E.; Chaturvedi, I.; Zhou, H.; Biswas, S.; and Huang, M. 2018. Augmenting end-to-end dialog systems with commonsense knowledge. In *AAAI*, 4970–4977.
- Zhang, R.; Lee, H.; Polymenakos, L.; and Radev, D. R. 2018a. Addressee and response selection in multi-party conversations with speaker interaction RNNs. In *AAAI*, 5690–5697.
- Zhang, S.; Dinan, E.; Urbanek, J.; Szlam, A.; Kiela, D.; and Weston, J. 2018b. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*, 2204–2213.
- Zhao, T.; Zhao, R.; and Eskenazi, M. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*, 654–664.
- Zheng, Y.; Chen, G.; Huang, M.; Liu, S.; and Zhu, X. 2019. Personalized dialogue generation with diversified traits. In *arXiv preprint arXiv:1901.09672*.
- Zheng, Y.; Zhang, R.; Huang, M.; and Mao, X. 2020. A pre-training based personalized dialogue generation model with persona-sparse data. In *AAAI*, 9693–9700.
- Zhou, H.; Huang, M.; Zhang, T.; Zhu, X.; and Liu, B. 2018a. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *AAAI*, 730–738.
- Zhou, H.; Young, T.; Huang, M.; Zhao, H.; Xu, J.; and Zhu, X. 2018b. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, 4623–4629.
- Zhou, X.; Dong, D.; Wu, H.; Zhao, S.; Yu, D.; Tian, H.; Liu, X.; and Yan, R. 2016. Multi-view response selection for human-computer conversation. In *EMNLP*, 372–381.