DAFA: Dialog System Domain Adaptation With a Filter and an Amplifier

JIANFENG YU, YAN YANG, CHENGCAI CHEN, LIANG HE, AND ZHOU YU

1Department of Computer Science, East China Normal University, Shanghai 200062, China
2Xiaoi Research, Shanghai 201803, China
3Department of Computer Science, University of California at Davis, Davis, CA 95616, USA

Corresponding author: Yan Yang (angel_yangyan@qq.com)

This work was supported by the Science and Technology Commission of Shanghai Municipality under Grant 18511105502, and in part by the Xiaoi Research.

ABSTRACT End-to-end task-oriented dialog systems have attracted vast amounts of attention in recent years, mainly because of their ease of training. However, such an end-to-end model requires a large number of labeled dialogs to train. Labeled dialogs are always difficult to obtain in real-world settings. We propose a domain adaptive end-to-end task-oriented dialog model that transfers knowledge in source domains to a target domain with limited training samples. Specifically, we design a domain adaptive filter in the encoder-decoder model to reduce useless features in source domains and preserve common features. A domain adaptive amplifier is designed to enhance the target domain impact. We evaluate our method on both synthetic dialog and human-human dialog datasets and achieve state-of-the-art results.

INDEX TERMS Neural network, dialogue system, domain adaptation, encoder-decoder model.

I. INTRODUCTION

Task-oriented dialog systems are widely used in daily life, which aims to achieve specific tasks through interactions between the system and users in natural language. They are typically developed for various specific domains, including hotel and restaurant searches [1], TV and laptop purchases [2], and flight reservations [3]. Each domain has its specific knowledge, therefore resulting in different dialog tasks. Most current end-to-end dialog systems are based on supervised learning and require sufficient training data. In reality, the data scarcity problem is very common with the increasing new needs in various dialog domains. For example, a company would not have enough data to train a product customer service system if the product was recently released. Therefore, it is important to develop learning methods that can utilize a small quantity of data to build effective dialog systems. We propose a Domain Adaptive dialog model with a Filter and an Amplifier (DAFA) to leverage resources from rich dialog domains to help build systems on domains with fewer data. DAFA learns general prior knowledge from rich-resource domains using a filter and then utilizes an amplifier to adapt to the low-resource target domain effectively.

Domain adaptation [4] can be considered one type of transfer learning method. Because every dialog domain is very different from each other, transferring knowledge from rich-resource domains to another low-resource domain is a challenging task. Few studies have proposed domain adaptive methods based on end-to-end dialog models [5]. DAFA is inspired by circuit design theory. There are filters and amplifiers in circuit design to reduce or enlarge signals. This idea has been applied to speech enhancement [6]. We can filter environmental noise first and then amplify the amplitude of a human voice. Following this idea, we propose to design a filter to first suppress source domain data signals that are not useful for the target domain and an amplifier to enlarge features that can improve target domain performance.

The filter utilizes a mask mechanism to retain useful information for decoding the belief span and generating a response in the source domain data. It is designed to acquire common features by a mask vector that filters some useless domain features in multiple source domains. The amplifier uses the self-attention mechanism to enlarge the impact of the target domain data. It is designed to gain more customized features in the target domain. Therefore, DAFA can effectively leverage useful information in the source domain and utilize a small quantity of data from the target domain. It achieves good performance on the movie booking task by only seeing one movie booking conversation.
We make the following contributions: (1) we propose a domain adaptation dialog system with a filter and an amplifier (DAFA) that can be applied to domains with a small quantity of training data. (2) A novel domain adaptation method in a dialog system is introduced to transfer knowledge in source domains to a target domain, which can filter out useless features in source domains and enhance useful features when adapting to the target domain. (3) Experiments on two task-oriented dialog system datasets show the superiority of the proposed method, which significantly outperforms state-of-the-art methods by more than 13.1% in the entity F1 score.

II. RELATED WORK
Recently, the end-to-end trainable dialog system framework has become popular due to its ease in training quality. Dialog system training is formalized as an encoder-decoder generation problem. Vinyals et al. used standard RNN and trained a task-oriented dialog system in a straightforward sequence-to-sequence (seq2seq) fashion [7], [8]. Task-oriented dialog systems are designed to achieve both task completion and human-like response generation. Lei et al. proposed a single seq2seq model, two-stage CopyNet (TSCP), that jointly optimizes belief state tracking and response generation [9]. This model significantly outperforms state-of-the-art methods on standard dialog datasets. However, training a good TSCP model on the Stanford multidomain dialog (KVRET) dataset requires more than a thousand dialogs. Reducing the available training data causes the TSCP model’s performance to decrease significantly. Therefore, we propose DAFA to extend TSCP to leverage rich source domain data to obtain a target domain system that only has limited training data.

Domain adaptation methods [4] involve two different types of datasets, one from a source domain and the other from a target domain. The source domain typically contains a sufficient quantity of annotated data to build a good dialog system, while there is often little or no labeled data in the target domain. Prior research has explored some generating tasks under the domain adaptation setting, such as image captioning and visual question answering [10], [11]. Walker et al. proposed a SPoT-based generator to address domain adaptation problems [12]. Subsequently, a system focused on tailoring user preferences [13] and controlling user perceptions of linguistic style [14] was proposed. Moreover, a phrase-based statistical generator [15] using graphical models and active learning and a multidomain procedure [16] via data counterfeiting and discriminative training was designed. However, domain adaptation for dialog systems has been less studied despite its important role in developing dialog models. Most research focuses on individual dialog model domain adaptation [17], [18], such as dialog state tracking. Recently, a zero-shot dialog generation [5] was proposed to enable the dialog model to generalize to unseen situations using minimal dialog data. Instead of zero-shot learning, we use a filter and an amplifier to train a domain adaptive dialog model. In summary, past domain adaptation research on dialog systems focused on the individual modules in a pipeline-based dialog system, while a small amount of research explored end-to-end trainable dialog systems.

III. PROBLEM FORMULATION
Task-oriented dialog models take a user utterance \( x \) as input and then generate the next response \( y \). We use the term \( d \) to describe the domain of training and testing data. Let \( D = D_s \cup D_t \) be a set of domains, where \( D_s \) is a set of source domains and \( D_t \) is a set of target domains. During training, we are given a sufficient set of samples \( \{x^{(n)}, y^{(n)}, d^{(n)}\} \sim p_{source}(x, y, d) \) drawn from source domains and a scarce set of samples \( \{x^{(m)}, y^{(m)}, d^{(m)}\} \sim p_{target}(x, y, d) \) drawn from the target domain. During testing, the model takes a user utterance \( x \) that belongs to dialogs in domain \( d \) from the target domains and generates the system response \( y \).

Our primary goal is to learn a generative dialog model \( M : X \times D \rightarrow Y \) that can perform well in a target domain by only using the scarce target domain data and the prior knowledge learned from rich source domains.

Train Data: \( \{x^{(n)}, y^{(n)}, d^{(n)}\} \sim p_{source}(x, y, d) \)
\( \{x^{(m)}, y^{(m)}, d^{(m)}\} \sim p_{target}(x, y, d) \)

Test Data: \( \{x, y, d\} \sim p_{target}(x, y, d) \)

Goal: \( M : X \times D \rightarrow Y \)
where \( n \) is sufficient, and \( m \) is scarce.

IV. PROPOSED METHOD
Our primary goal is to transfer the task-oriented dialog model from source domains to target domains. We build DAFA based on TSCP [9], a single sequence-to-sequence (seq2seq) model with two-stage CopyNet that separately decodes the belief span (bspans) and the machine response. The bspan is used in TSCP to track the dialog state. In TSCP, it records the dialog state in a text span to be decoded by the model. For example, it has an information field (marked with <Inf>) Italian; cheap <Inf> y. We insert a filter and an amplifier in the TSCP model to achieve domain adaptation. Figure 1 shows the model overview. We use a filter to decrease useless features from source domains and an amplifier to obtain customized features from the target domain. Thus, the model can learn both common knowledge and customized knowledge. It can perform well, even though the target domain data are insufficient.

An encoder is designed to encode bspan \( b \), response \( y \) and user utterance \( x \). Training data will be filtered if it is from the source domains. The filter first calculates the mask vector \( m \) and then multiplies \( m \) and the hidden state \( h \) to obtain \( h' \). After that, it goes to the decoder, which is divided into two steps. Similar to TSCP, we denote the system’s belief state as bspans. The first step decodes bspans, and the second step generates the response by using bspans. However, if the training data come from the target domain, the data will go to the amplifier between the encoder and the decoder. The amplifier uses a self-attention mechanism to calculate the attention score matrix of the hidden state \( h \). After the softmax
Algorithm 1 Domain Adaptive Training

Initialize weights of DAFA;
Data = \( \{x^{(n)}, y^{(n)}, d^{(n)}\}_{source}, \{x^{(m)}, y^{(m)}, d^{(m)}\}_{target} \);
1: while batch \( \sim \{x^{(n)}, y^{(n)}, d^{(n)}\}_{source} \) do
2: Generate bspans
3: Generate responses
4: Backpropagate loss \( L_{\text{filter}} \)
5: end while
6: while batch \( \sim \{x^{(m)}, y^{(m)}, d^{(m)}\}_{target} \) do
7: Generate bspans
8: Generate responses
9: Backpropagate loss \( L_{\text{amplifier}} \)
10: end while

operates on the attention score matrix, it is multiplied with the hidden state \( h \) to obtain \( h' \), and then it goes to the decoder. Algorithm 1 shows the method’s training procedure. If the training data are from the source domain batch, it goes to filter. Otherwise, it goes to the amplifier. We describe each component in detail as follows:

**A. ENCODER**

We use a simple dynamic encoder to encode the dialog context, which is a 1-layer bidirectional GRU. Given a source sequence of tokens \( X = x_1, x_2, \ldots, x_n \), an encoder network represents \( X \) as hidden states: \( \mathbf{H}^{(x)} = h_1^{(x)}, h_2^{(x)}, \ldots, h_n^{(x)} \). The input \( x \) is a pretrained GloVe word embedding sequence of \( X \).

Specifically, the hidden state \( h_i^{(x)} \) is calculated as follows:

\[
\begin{align*}
z_i &= \sigma(W_z x_i + U_i h_{i-1}) \quad (1) \\
\tilde{r}_i &= \sigma(W_r x_i + U_r h_{i-1}) \quad (2) \\
h_i^{(x)} &= \tanh(W x_i + U_i \cdot h_i^{(x)}) \quad (3) \\
\hat{h}_i^{(x)} &= (1 - z_i) \cdot h_{i-1}^{(x)} + z_i \cdot h_i^{(x)} \quad (4)
\end{align*}
\]

where \( W_z, W_r, U_z, U_r, \) and \( U \) are parameters learned, \( \sigma \) is an elementwise sigmoid nonlinearity, \( z_i \) is the update gate in GRU, and \( \tilde{r}_i \) is the reset gate in GRU.

**B. FILTER FOR SOURCE DATA**

The output of the encoder then feeds into the filter module if the data are from source domains. The filter is designed with a mask mechanism to decrease useless features in source domains. This enhances the dialog model’s capacity to acquire common features in the source domain. Because our source domain data have multiple domains, the model can learn common knowledge in these data. Here, common knowledge means common features in source domains that we want to transfer to a target domain. Because these common features are useful in several source domains, they may also be useful in the target domain. In source domains, our primary goal is to acquire more common knowledge. However, if we do not have a filter, some useless features are encoded in the hidden state from the source domain data, and they are not useful in the target domain training. Therefore, these useless features have a bad effect on the model’s capacity to learn common knowledge, which is similar to environmental noise in electronic signals. In the training phase, the filter can help
the model gain common features by filtering useless features. As the domain is different between the source and target data, common knowledge of the model learned in source domains is useful in the target domain.

We introduce a mask vector $m_{\text{domain}}$, which decides which parts of the hidden state $h$ should be filtered. $m_{\text{domain}}$ is learned as a function of $h^{(x)}_i$ and $h^{(y)}_j$ so that it can filter noisy information in hidden states.

Specifically, the filter first calculates the mask vector $m_{\text{domain}}$ according to $h^{(x)}_i$ and $h^{(y)}_j$, where $h^{(x)}_i$ is the hidden state of the encoder’s output in $t_i$, and $h^{(y)}_j$ is the hidden state of the decoder in $t_j$. Then, it performs elementwise multiplication $m_{\text{domain}}$ times $h^{(x)}_i$ and $h^{(y)}_J$. Finally, the filter outputs $h^{(x)}_i$ and $h^{(y)}_J$. They are sent to the decoder for calculating attention to the weight and decoding $y_t$.

$$m_{\text{domain}}(ij) = \sigma(W_x h^{(x)}_i + W_y h^{(y)}_j + b_1) \quad (5)$$

$$h^{(x)}_i = m_{\text{domain}}(ij) \odot h^{(x)}_i \quad (6)$$

$$h^{(y)}_j = m_{\text{domain}}(ij) \odot h^{(y)}_j \quad (7)$$

where $W_x$, $W_y$ and $b_1$ are parameters learned, and $\sigma$ is an elementwise sigmoid nonlinearity.

C. AMPLIFIER FOR TARGET DATA

To reduce data dependence in the target domain, we use a domain amplifier to help the model acquire more target domain customized features in training. Since the domain adaptive dialog model is only tested in the target domain, gaining more customized features would make the model perform better in the target domain. The amplifier is implemented using a self-attention mechanism [19], which can increase the weights of customized features in the target domain. Self-attention can be used in NLP tasks to calculate the relationship between two tokens and to learn the internal structure of a sentence. Here, we use the amplifier to capture dependencies between two words and the features of a sentence in the target domain.

More precisely, the calculation process is shown in figure 2. The amplifier first calculates the similarity function $f_{ij}(h^{(x)}_i, h^{(y)}_j)$ (Eq. 8) between all $h^{(x)}_i \in H^{(x)}$ and $h^{(y)}_j \in H^{(y)}$. Then, the similarity score matrix $F$ and the matrix $H_{\text{Copy}}$ copied by $H^{(x)}$ are used to calculate $H^{(x)}$ in Eq (9).

In addition, $g(\cdot)$ is used to sum along rows, and $\text{softmax}(\cdot)$ means softmax along rows.

$$f_{ij}(h^{(x)}_i, h^{(y)}_j) = c \cdot \tanh([W_1 h^{(x)}_i + W_2 h^{(y)}_j + b_2]/c) + M_{ij} \cdot 1 \quad (8)$$

$$H^{(x)} = g(\text{softmax}(F \otimes H_{\text{Copy}})) \quad (9)$$

where $W_1 \in \mathbb{R}^{d \times d}$, $W_2 \in \mathbb{R}^{d \times d}$ and $b_2 \in \mathbb{R}^d$ are parameters trained on target domain training data from random initialization, $d$ is the embedding size, $c$ is a scalar, 1 is an all-one vector, and $\otimes$ is an elementwise product.

$$M_{ij} = \begin{cases} 0, & i \neq j \\ -\infty, & i = j \end{cases} \quad (10)$$

D. DECODER

In the TSCP model, Lei et al. proposed that foregoing solutions of belief tracking can be updated by applying seq2seq models directly to the problem [9]. In contrast to the research of Wen et al. [20], which treated slot values as classification labels, and Lei et al. recorded them in a text span to be decoded by the model [9]. This leverages the state-of-the-art neural seq2seq models to learn and dynamically generate them. Specifically, at turn $t$, the model only needs to refer to $b_{t-1}$, $y_{t-1}$ and $x_t$ to generate a new bspan $b_t$ and machine response $y_t$, without appealing to knowing all past utterances. This Markov assumption allows the TSCP model to concatenate $b_{t-1}$, $y_{t-1}$ and $x_t$ (denoted as $b_{t-1} y_{t-1} x_t$) as a source sequence for seq2seq modeling to generate $b_t$ and $y_t$ as target output sequences at each turn. $b_t$ and $y_t$ are processed separately, as the belief state $b_t$ only depends on $b_{t-1} y_{t-1} x_t$, while the response $y_t$ is additionally conditioned on $b_t$ and the knowledge base search results.

For example, at the first turn, a user says, “Can I have some Italian food please?” Thus, $b_1$ contains an information slot “Italian.” During the second turn, the user adds an additional constraint cheap in $x_2$, resulting in two slot values in $b_2$’s information field. In the third turn, the user further asks for the restaurant’s phone and address, which are stored in the requested slots of $b_3$.

We use $h^{(x)}_i \in H^{(x)}$ and $h^{(y)}_j$ to represent the hidden states because the filter and the amplifier have different calculations. If the decoder obtains the output of the filter, $h^{(x)}_i$ and $h^{(y)}_j$ equal $h^{(x)}_i$ (Eq. 6) and $h^{(y)}_j$ (Eq. 7). Otherwise, $h^{(x)}_i$ is equal to $h^{(x)}_i \in H^{(x)}$ (Eq. 9), and $h^{(y)}_j$ is the original $h^{(y)}_j$. Based on the output by the filter and amplifier, a decoder network generates a target sequence of tokens $Y = y_1, y_2, \ldots, y_m$ whose likelihood should be maximized given the training corpus. For decoding $y_t$, the decoder takes the hidden vector $h^{(y)}_j$ and then calculates the attention scores between all $h^{(x)}_i \in H^{(x)}$ and $h^{(y)}_j$. After that, it sums all $h^{(x)}_i$, weighted by their corresponding attention scores.
A good dialog system needs to complete tasks effectively and generate natural responses that are easy for users to understand. Therefore, we assess the effectiveness of DAFA in two aspects: the task success rate and language quality. The evaluation metrics are listed as follows:

- **BLEU** evaluates the language quality [21] of generated responses (hence the top-1 candidate in [20]).

  \[ \text{BLEU} = 1^{T} \tanh(W_3h_i^{(x)} + W_4h_j^{(y)}) \]  

- **Entity** \( F_1 \) determines if a generated response contains the correct entities (slots) in the reference response.

  \[ \text{Entity} \quad F_1 \text{ determines if a generated response contains the correct entities (slots) in the reference response.} \]

The summed result \( h_j^{(x)} \) concatenates \( h_j^{(y)} \) as a single vector that is mapped into an output space for a softmax operation to decode the current token.

\[
\begin{align*}
    u_{ij} &= v^T \tanh(W_3h_i^{(x)} + W_4h_j^{(y)}) \\
    h_j^{(x)} &= \sum_{i=1}^{n} \frac{e^{u_{ij}}}{\sum_i e^{u_{ij}}} h_i^{(x)} \\
    y_j &= \text{softmax}(O \begin{bmatrix} h_j^{(x)} \\ h_j^{(y)} \end{bmatrix})
\end{align*}
\]

where \( v, W_3 \) and \( W_4 \) are learned parameters.

**V. EXPERIMENTS AND RESULTS**

A good dialog system needs to complete tasks effectively and generate natural responses that are easy for users to understand. Therefore, we assess the effectiveness of DAFA in two aspects: the task success rate and language quality. The evaluation metrics are listed as follows:

- **BLEU** evaluates the language quality [21] of generated responses (hence the top-1 candidate in [20]).

- **Entity** \( F_1 \) determines if a generated response contains the correct entities (slots) in the reference response.

**A. DATASETS**

The statistics of the datasets are shown in Table 1. SimDial, developed by Zhao and Eskenazi [5], is a multidomain dialog generator that can generate realistic conversations for slot-filling domains with configurable complexity. Compared to other synthetic dialog corpora used to test generative end-to-end dialog models, e.g., bAbI [22], SimDial data are significantly more challenging. First, since SimDial simulates communication noise, the dialogs that are generated can be very long (more than 50 turns), and the simulated agent can carry out error recovery strategies to correctly infer the users’ goals. This challenges end-to-end models to model long dialog contexts. Second, SimDial simulates spoken language phenomena, e.g., self-repair and hesitation. Prior work [23] has shown that this type of utterance-level noise deteriorates end-to-end dialog system performance.

SimDial is a simulated dialog dataset with six domains: restaurant, movie, bus, restaurant-slot, restaurant-style and weather. For each domain, 900, 100, and 500 dialogs were generated for training, validation and testing, respectively. On average, each dialog has 26 utterances, and each utterance has 12.8 word tokens. The total vocabulary size is 651. For a fair comparison, we use the evaluation setup from Zhao et al. [5]. We split the dataset such that the training data include dialogs from the restaurant, bus and weather domains and the test data include the restaurant, restaurant-slot, restaurant-style and movie domains.

- **Restaurant (in domain):** evaluation on the restaurant test data checks if a dialog model can maintain its performance on source domains because we want to see whether our model is valid.

- **Restaurant-slot (unseen slots):** restaurant-slot has the same slot types and natural language generation (NLG) templates as the restaurant domain but has a completely different slot vocabulary, i.e., different location names and cuisine types. Thus, this is designed to evaluate a model that can generalize to unseen slot values.

- **Restaurant-style (unseen NLG):** restaurant-style has the same slot type and vocabulary as restaurant, but its NLG templates are completely different, e.g., “which cuisine type? “→ “please tell me what kind of food you prefer”. This part tests whether a model can learn to adapt to generate novel utterances with similar semantics.

- **Movie (new domain):** movie has completely different NLG templates and structure and shares few common traits with the source domains at the surface level. The movie is the hardest task in SimDial data, which challenges a model to correctly generate the next responses that are semantically different from those in source domains.

The second dataset we experiment on is the Stanford multidomain dialog (KVRET) dataset [24]. It has 3,031 human-dial corpus train (in domain) and 1,601. We experiment on KVRET to validate whether our proposed method generalizes to human-human dialogs.

**B. EXPERIMENTAL SETTINGS**

We set the DAFA’s hidden state size and the embedding size to 50. We train the model with an Adam optimizer [25], with a learning rate of 0.003 for supervised training. Early stopping is performed on the developing set. In generation, we use beam search for decoding, with a beam size of 10. For the training procedure, we first train DAFA with dialogs from source domains to convergence and then feed the target domain training dialogs to fine-tune the model.
TABLE 2. Results evaluated on SimDial by comparing our model with base models (use 1% training data).

| In domain     | ZSDG   | Transfer | DAFA  | Transfer-oneshot | DAFA-oneshot | DAFA\filter  | DAFA\amp     |
|---------------|--------|----------|-------|------------------|---------------|---------------|---------------|
| BLEU          | 70.1   | 51.8     | 51.7  | 51.1             | 51.3          | 51.5          | 51.3          |
| Entity F1     | 79.9   | 88.5     | **90.3** | 87.6            | 90.1          | 89.9          | 87.7          |
| Unseen Slot   | ZSDG   | Transfer | DAFA  | Transfer-oneshot | DAFA-oneshot  | DAFA\filter  | DAFA\amp     |
| BLEU          | 68.5   | 43.3     | 40.6  | 40.8             | 40.2          | 40.1          | 39.7          |
| Entity F1     | 74.6   | 78.7     | 74.9  | 70.1             | 72.2          | 79.2          | 73.1          |
| Unseen NLG    | ZSDG   | Transfer | DAFA  | Transfer-oneshot | DAFA-oneshot  | DAFA\filter  | DAFA\amp     |
| BLEU          | 70.1   | 30.6     | 22.1  | 20.0             | 19.7          | 21.7          | 21.9          |
| Entity F1     | 72.9   | 82.2     | 76.7  | 82.8             | 69.7          | **83.8**      | 74.8          |

**C. BASELINES**

The domain adaptation method for task-oriented dialog systems has been less studied. Therefore, we compared the proposed model with some similar models. In this paper, we use three baselines for comparison:

- **ZSDG** [5] is the state-of-the-art dialog domain adaptation model. This model strengthens the LSTM-based encoder-decoder with an action matching mechanism. The model samples 100 labeled utterances as domain description seeds for domain adaptation.
- **TSCP** [9] utilizes a two-stage CopyNet to track what dialog believes and generate responses. TSCP is not given any source domain information. We want to use this baseline to understand the effectiveness of incorporating the source domain data in training.
- **Transfer learning** [26], [27]. We apply transfer learning on the TSCP model as the third baseline. We first pretrain the model by mixing data from source domains and then fine-tune on the target domain. We also enlarge the vocabulary with the training data in the target domain. In addition, we implement a one-shot learning version of this model by only using one target domain dialog for adaptation, as a comparison with the one-shot learning case of DAFA.

The experimental settings of transfer one-shot and DAFA one-shot are the same as those of the transfer model and DAFA model except for the amount of target domain data. One-shot has only one sample to train in the target domain.

We also performed an ablation study to examine the effectiveness of the filter and the amplifier.

- DAFA\filter. We removed the filter in DAFA.
- DAFA\amp. We removed the amplifier in DAFA.

**D. RESULTS**

Table 2 shows the results on baselines and our model evaluated on SimDial data. We ran the experiments five times to average. Nine dialogs, which are 1% of the source domain training data, are used as the target domain to train the transfer, DAFA, DAFA\filter and DAFA\amp models. One target dialog is used to train transfer one-shot and DAFA one-shot models. ZSDG [5] only uses domain descriptions in the target domain. We find that DAFA one-shot performs better than transfer one-shot and ZSDG in the new domain (movie) in terms of both BLEU and Entity F1. This suggests that the filter and amplifier in DAFA are useful in adapting to new domains. If we increase the target domain data from one dialog to nine dialogs, DAFA still outperforms the transfer learning models on the new domain. DAFA with nine dialogs achieves the best performance on the new domain with 65.7% in Entity F1. If we remove either the filter or the amplifier in DAFA, the performance on the new domain decreases. However, they are still better than the transfer learning model. This suggests that only having the filter or the amplifier still improves the adaptation results on the new domain compared to the vanilla transfer model. We can also find that ZSDG has 54.6 in BLEU for the new domain, which is very high. This is because ZSDG uses an utterance GRU and a discourse LSTM to encode the entire dialog context, while transfer and DAFA are based on TSCP, which only encodes the current user utterance, last turn’s bspan and response. Therefore, ZSDG has outstanding language quality, but it needs a longer time to train.

However, DAFA does not have the best performance in “unseen slot” and “unseen NLG”. Since these two domains are generated from one of the source domains (restaurant domain), they still share some restaurant domain features with the original domain data, although the slot vocabulary and NLG templates are completely different. Our model DAFA has a filter in source domain training. We design it to weaken source domain features, so the model can learn general knowledge. However, in these two domains, the filter has negative effects on model performance because it weakens restaurant domain features that are useful and cannot be acquired in the target domain. This also explains why DAFA still performs well in “in domain”, because attenuated restaurant domain features can still be acquired in the target domain. Ablation studies also illustrate this problem. For the “unseen slot” and “unseen NLG”, DAFA without a filter can obtain the best results among all models. This means that when the target domain is related to a domain in the source, the filter may be harmful to the
TABLE 3. An example dialog generated by DAFA and transfer.

| Turn | User                                      | Response(Transfer)            | Response(DAFA)                  |
|------|-------------------------------------------|-------------------------------|---------------------------------|
| 1    | where can I find a parking garage?        | you’re welcome                | [poi_SLOT] is [distance_SLOT] away. |
| 2    | [poi_SLOT] is at 481 Amaranta ave.         | you are a car collision nearby.| I have a car collision nearby, I’m here to you have a road-block nearby. |
| 3    | I will send on your screen the quickest route, sure, there is heavy traffic now ... thanks for the notification. | you are your welcome.          | you’re welcome.                  |

TABLE 4. Results evaluated on KVRET by comparing our models with base models. TSCP(full) uses full target domain training data and no source domain data. TSCP(1%) uses 1% target domain training data and no source domain data. Others use source domain data, and 1% target domain training data (ZSDG uses no target domain training data).

| Scheduling | BLEU  | Entity F1 | TSCP(full) | TSCP(1%) | ZSDG | Transfer | DAFA |
|------------|-------|-----------|------------|----------|------|----------|------|
| Weather    |       |           | 21.5       | 8.1      | 18.1 | 31.0     | 40.3 |
| Navigation |       |           | 19.4       | 5.9      | 14.9 | 19.9     | 20.5 |

We also investigate the impact of using different quantities of target domain data on model performance. We use the model trained on KVRET data with weather and navigation as source domains and scheduling as the target domain. Figure 3 shows that the system’s performance positively correlates with the amount of target training data available. Both entity F1 and BLEU scores nearly converge when 27 (3% of data) dialogs are used.

E. ANALYSIS

We first compare DAFA with transfer in Table 3 to show that our model DAFA demonstrates great performance in dialog domain adaptation. We use an example dialog from KVRET’s navigation task when it is treated as the target domain. For the user utterance “where can I find a parking garage?”, both transfer and DAFA generate the bspan “parking garage” successfully; however, transfer fails to generate the response. The transfer generates “you’re welcome”, a greeting response, which appears frequently in training data. However, DAFA can give the response “[poi SLOT] is [distance SLOT] away”. Obviously, it can be seen that DAFA can adapt to a new domain better.

DAFA learns the shared knowledge through the filter and the amplifier, which helps generate responses in a new target domain. The filter can learn more common knowledge by masking out the features that are not useful and only keep the useful features in generation during source data training.
The amplifier can improve the model performance on novel utterances in the target domain. The amplifier can catch more customized features in the target domain so that it can help the model learn more new knowledge by enhancing the features of the target domain. We also conducted a qualitative analysis to compare responses generated from DAFA, DAFA\{filter\} and DAFA\{amp\}. A general utterance, e.g., “See you next time”, which appears in all domains, can be correctly generated by all three models. However, utterances with unseen slots are different, for example, explicitly confirming “Do you mean [food_type]?”. DAFA\{amp\} fails in this situation since the new slot values are not in its vocabulary. While DAFA still performs well since it learns to copy entity-like words from the context, the overall generated sentence is often incorrect, e.g., “Do you mean romance food?”. For unseen templates, both the DAFA\{filter\} and DAFA\{amp\} can only generate correct dialog acts but incorrect dialog responses. Only DAFA, with both a filter and an amplifier, can infer that sentences “[movie_name] is a great movie” and “[bus_name] can take you there” are both “provide information”. By leveraging the learned dialog act, DAFA can generate a more appropriate response.

VI. CONCLUSION AND FUTURE WORK

In this paper, we present DAFA, a new domain adaptive end-to-end trainable dialog model. DAFA is a seq2seq model that injects a filter for source domain data and an amplifier for target domain data. Specifically, DAFA uses a domain adaptive filter to attenuate information that is not necessary for the source domain and a domain adaptive amplifier to emphasize target domain data to achieve good performance on the target domain that has limited data. We tested DAFA on both simulated conversations and human-human conversations. Experiments show that DAFA outperforms other domain adaptation models in the new, unseen, low-resource domain. In the future, we will generalize DAFA to work on other NLP tasks that use encoder-decoder models, such as machine translation.

REFERENCES

[1] T.-H. Wen, M. Gasic, D. Kim, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, “Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking,” in Proc. 16th Annu. Meeting Special Interest Group Discourse Dialogue, 2015, p. 275.

[2] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, “Semantically conditioned LSTM-based natural language generation for spoken dialogue systems,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 1711–1721.

[3] E. Levin, S. Narayanan, R. Pieraccini, K. Biatov, E. Bocchiere, G. D. Fabbrizio, W. Eckert, S. Lee, A. Pokrovsky, and M. Rahim, “The AT&T-DARPA communicator mixed-initiative spoken dialog system,” in Proc. 6th Int. Conf. Spoken Lang. Process., 2000, pp. 1–4.

[4] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, “Analysis of representations for domain adaptation,” in Proc. Adv. Neural Inf. Process. Syst., 2007, pp. 137–144.

[5] T. Zhao and M. Eskenazi, “Zero-shot dialog generation with cross-domain latent actions,” 2018, arXiv:1805.04803. [Online]. Available: http://arxiv.org/abs/1805.04803

[6] Y. Tsao and Y.-H. Lai, “Generalized maximum a posteriori spectral amplitude estimation for speech enhancement,” Speech Commun., vol. 76, pp. 112–126, Feb. 2016.

[7] O. Vinyals and Q. Le, “A neural conversational model,” 2015, arXiv:1506.05869. [Online]. Available: http://arxiv.org/abs/1506.05869

[8] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 3104–3112.

[9] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. Yin, “Secuicity: Simplifying task-oriented dialogue systems with single Sequence-to-Sequence architectures,” in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Long Papers), vol. 1, 2018, pp. 1437–1447.

[10] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell, “Multi-content GAN for few-shot font style transfer,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7564–7573.

[11] X. Dong, L. Zhu, D. Zhang, Y. Yang, and F. Wu, “Fast parameter adaptation for few-shot image captioning and visual question answering,” in Proc. ACM Multimedia Conf. Multimedia Conf. (MM). New York, NY, USA: ACM, 2018, pp. 54–62.

[12] M. A. Walker, O. Rambow, and M. Rogati, “SPOiT: A trainable sentence planner,” in Proc. 2nd Meeting North Amer. Chapter Assoc. Comput. Linguistics Lang. Technol. (NAACL). Stroudsburg, PA, USA: Association for Computational Linguistics, 2001, pp. 1–8.

[13] M. A. Walker, A. Stent, F. Mairese, and R. Prasad, “Individual and domain adaptation in sentence planning for dialogue,” J. Artif. Intell. Res., vol. 30, pp. 413–456, Nov. 2007.

[14] F. Mairese and M. A. Walker, “Controlling user perceptions of linguistic style: Trainable generation of personality traits,” Comput. Linguistics, vol. 37, no. 3, pp. 455–488, Sep. 2011.

[15] F. Mairese, M. Gašić, F. Juričček, S. Keizer, B. Thomson, K. Yu, and S. Young, “Phrase-based statistical language generation using graphical models and active learning,” in Proc. 45th Annu. Meeting Assoc. Comput. Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 1552–1561.

[16] T.-H. Wen, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, and S. Young, “Multi-domain neural network language generation for spoken dialogue systems,” 2016, arXiv:1603.01232. [Online]. Available: http://arxiv.org/abs/1603.01232

[17] N. Mrksic, D. Ř Seághadha, B. Thomson, M. Gasic, P.-H. Su, D. Vandyke, and T.-H. Wen, “Multi-domain dialog state tracking using recurrent neural networks,” in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. (Short Papers), vol. 2, 2015, pp. 794–799.

[18] J. Williams, “Multi-domain learning and generalization in dialog state tracking,” in Proc. SIGDIAL Conf., 2013, pp. 433–441.

[19] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, “Disan: Directional self-attention network for rnn/cnn-free language understanding,” in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 5446–5455.

[20] T.-H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, “A network-based end-to-end trainable task-oriented dialogue system,” 2016, arXiv:1604.04562. [Online]. Available: http://arxiv.org/abs/1604.04562

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLUE: A method for automatic evaluation of machine translation,” in Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL). Stroudsburg, PA, USA: Association for Computational Linguistics, 2001, pp. 311–318.

[22] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, and J. Weston, “Evaluating prerequisite qualities for learning end-to-end dialog systems,” 2015, arXiv:1511.06931. [Online]. Available: http://arxiv.org/abs/1511.06931

[23] A. Eshghi, I. Shalyminov, and O. Lemon, “Bootstrapping incremental dialogue systems from minimal data: The generalisation power of dialogue grammars,” 2017, arXiv:1709.07858.

[24] M. Bizzarri and C. D. Manning, “Key-value retrieval networks for task-oriented dialogue,” 2017, arXiv:1705.05414. [Online]. Available: http://arxiv.org/abs/1705.05414

[25] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980. [Online]. Available: https://arxiv.org/abs/1412.6980

[26] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Long Papers), vol. 1, 2018, pp. 328–339.
JIANFENG YU was born in Hangzhou, China, in 1995. He received the B.E. degree from the Department of Computer Science and Technology, China University of Mining and Technology, in 2017. He is currently pursuing the master’s degree with the Department of Computer Science and Technology, East China Normal University. His research interests include knowledge analysis and data processing.

YAN YANG received the M.Sc. and Ph.D. degrees from East China Normal University. She is an Assistant Professor with the Department of Computer Science and Technology, East China Normal University, China. She holds five patents, and she published more than ten research articles in international conferences and journals. Her research interests include information extraction, dialogue systems, and knowledge processing. She received the Shanghai Science and Technology Progress Award and won second prize, in 2016.

CHENGCAI CHEN received the B.Eng. degree in industrial analysis from the Department of Applied Chemistry, Qingdao University of Science and Technology. He is currently the President of Xiaoai Research and the Vice President of Technology of Xiaoai Robots. His research interests include QA systems and knowledge base, NLP, data-driven machine learning, and knowledge representation and reasoning.

LIANG HE received the bachelor’s and Ph.D. degrees from the Department of Computer Science and Technology, East China Normal University, China. He is currently a Professor and the Director of the Department of Computer Science and Technology, East China Normal University. His current research interests include knowledge processing, user behavior analysis, and context-aware computing. He has been awarded the Star of Talent in Shanghai. He is also a Council Member of the Shanghai Computer Society, a member of the Academic Committee, the Director of the Technical Committee of Shanghai Engineering Research Center of Intelligent Service Robots, and a Technology Foresight Expert of Shanghai Science and Technology in his research areas. In recent years, he has hosted a number of national science and technology supports and participated at the National 13th Five-Year Technology Support Program of Shanghai Science and Technology Long-Term Development Plan, and Shanghai 13th Five-Year Science and Technology Program. He has received the Shanghai Science and Technology Progress Award five times, and won first-prize, in 2013, and second prize, in 2015. He holds more than ten patents, and he has published two monographs and more than 70 refereed articles in national and international journals and conference proceedings.

ZHOU YU received the B.S. degree in computer science and the B.A. degree in linguistics in the English language from Zhejiang University, in 2011, and the Ph.D. degree from the School of Computer Science, Language Technology Institute, Carnegie Mellon University, in 2017. She is an Assistant Professor of computer science with the University of California at Davis, Davis. She is the Director of the Davis NLP Lab. She designs algorithms for real-time intelligent interactive systems that coordinate with user actions that are beyond spoken languages, including nonverbal behaviors to achieve effective and natural communications. She is interested in building robust and multipurpose dialog systems using fewer data points and less annotation. She is also passionate about language generation. Her work Persuasion for Good recently received an ACL 2019 best paper nomination. She was featured in Forbes as 2018’s 30 under 30 in science for her work on multimodal dialog systems. Her team recently won the 2018 Amazon Alexa Prize on building an engaging social bot for a U.S. $500,000 award.

* * *