

Dynamic Fusion Network for Multi-Domain End-to-end Task-Oriented Dialog

Libo Qin[†], Xiao Xu[†], Wanxiang Che^{†*}, Yue Zhang[‡], Ting Liu[†]

[†]Research Center for Social Computing and Information Retrieval

[†]Harbin Institute of Technology, China

[‡]School of Engineering, Westlake University, China

[‡]Institute of Advanced Technology, Westlake Institute for Advanced Study

[†]{lbqin, xxu, car, tliu}@ir.hit.edu.cn, yue.zhang@wias.org.cn

Abstract

Recent studies have shown remarkable success in end-to-end task-oriented dialog system. However, most neural models rely on large training data, which are only available for a certain number of task domains, such as navigation and scheduling. This makes it difficult to scalable for a new domain with limited labeled data. However, there has been relatively little research on how to effectively use data from all domains to improve the performance of each domain and also unseen domains. To this end, we investigate methods that can make explicit use of domain knowledge and introduce a shared-private network to learn shared and specific knowledge. In addition, we propose a novel Dynamic Fusion Network (DF-Net) which automatically exploit the relevance between the target domain and each domain. Results show that our model outperforms existing methods on multi-domain dialogue, giving the state-of-the-art in the literature. Besides, with little training data, we show its transferability by outperforming prior best model by 13.9% on average.

1 Introduction

Task-oriented dialogue systems (Young et al., 2013) help users to achieve specific goals such as restaurant reservation or navigation inquiry. In recent years, end-to-end methods in the literature usually take the sequence-to-sequence (Seq2Seq) model to generate a response from a dialogue history (Eric and Manning, 2017; Eric et al., 2017; Madotto et al., 2018; Wen et al., 2018; Gangi Reddy et al., 2019; Qin et al., 2019b; Wu et al., 2019a). Taking the dialogue in Figure 1 as an example, to answer the drivers query about the “gas station”, the end-to-end dialogue system directly generates system response given the query and a corresponding knowledge base (KB).

*Email corresponding.

Knowledge Base (KB)

Address	Distance	POI type	POI	Traffic info
5672 barringer street	5 miles	certain address	5672 barringer street	no traffic
200 Alester Ave	2 miles	gas station	Valero	road block nearby
899 Ames Ct	5 miles	hospital	Stanford Childrens Health	moderate traffic
481 Amaranta Ave	1 miles	parking garage	Palo Alto Garage R	moderate traffic

Dialogue

Driver	Address to the gas station.
Car	Valero is located at 200 Alester Ave.
Driver	OK, please give me directions via a route that avoids all heavy traffic.
Car	Since there is a road block nearby, I found another route for you and I sent it on your screen.

Figure 1: Example of a task-oriented dialogue that incorporates a knowledge base (KB) from the SMD dataset (Eric et al., 2017). Words with the same color refers queried entity from the KB. Better viewed in color.

Though achieving promising performance, end-to-end models rely on a considerable amount of labeled data, which limits their usefulness for new and extended domains. In practice, we cannot collect rich datasets for each new domain. Hence, it is important to consider methods that can effectively transfer knowledge from a source domain with sufficient labeled data to a target domain with limited or little labeled data.

Existing work can be classified into two main categories. As shown in Figure 2(a), the first strand of work (Eric and Manning, 2017; Eric et al., 2017; Madotto et al., 2018; Wu et al., 2019a) simply combines multi-domain datasets for training. Such methods can implicitly extract the shared features but fail to effectively capture domain-specific knowledge. As shown in Figure 2(b), The second strand of work (Wen et al., 2018; Qin et al., 2019b) trains model separately for each domain, which can better capture domain-specific features. However, those methods ignore shared knowledge between different domains (e.g. the location word exists in both schedule domain and navigation domain).

We consider addressing the limitation of existing work by modeling knowledge connections between domains explicitly. In particular, a simple baseline

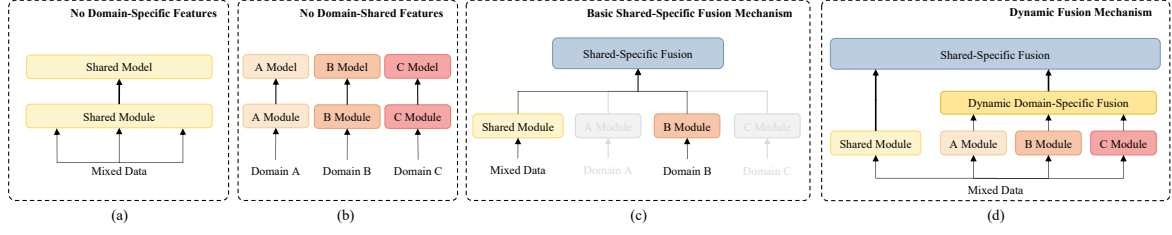


Figure 2: Methods for multi-domain dialogue. Previous work either trains a general model on mixed multi-domain mixed datasets (a), or on each domain separately (b). The basic shared-private framework is shown (c). Our proposed extension with dynamic fusion mechanism is shown (d).

to incorporate domain-shared and domain-private features is shared-private framework (Liu et al., 2017; Zhong et al., 2018; Wu et al., 2019b). Shown in Figure 2(c), it includes a shared module to capture domain-shared feature and a private module for each domain. The method explicitly differentiates shared and private knowledge. However, this framework still has two issues: (1) given a new domain with extremely little data, the private module can fail to effectively extract the corresponding domain knowledge. (2) the framework neglects the fine-grained relevance across certain subsets of domains. (e.g. schedule domain is more relevant to the navigation than to the weather domain.)

To address the above issues, we further propose a novel Dynamic Fusion Network (DF-Net), which is shown in Figure 2 (d). In contrast to the shared-private model, a dynamic fusion module (see §2.3) is further introduced to explicitly capture the correlation between domains. In particular, a gate is leveraged to automatically find the correlation between a current input and all domain-specific models, so that a weight can be assigned to each domain for extracting knowledge. Such a mechanism is adopted for both the encoder and the decoder, and also a memory module to query knowledge base features. Given a new domain with little or no training data, our model can still make the best use of existing domains, which cannot be achieved by the baseline model.

We conduct experiments on two public benchmarks, namely SMD (Eric et al., 2017) and MultiWOZ 2.1 (Budzianowski et al., 2018). Results show that our framework consistently and significantly outperforms the current state-of-the-art methods. With limited training data, our framework outperforms the prior best methods by 13.9% on average.

To our best of knowledge, this is the first work to effectively explore shared-private framework in

multi-domain end-to-end task-oriented dialog. In addition, when given a new domain which with few or zero shot data, our extended dynamic fusion framework can utilize fine-grained knowledge to obtain desirable accuracies, which makes it more adaptable to new domains.

All datasets and code are publicly available at: <https://github.com/LooperXX/DF-Net>.

2 Model Architecture

We build our model based on a seq2seq dialogue generation model (§2.1), as shown in Figure 3(a). To explicitly integrate domain awareness, as shown in Figure 3(b) we first propose to use a shared-private framework (§2.2) to learn shared and the corresponding domain-specific features. Next, we further use a dynamic fusion network (§2.3) to dynamically exploit the correlation between all domains for fine-grained knowledge transfer, which is shown in Figure 3(c). In addition, adversarial training is applied to encourage shared module generate domain-shared feature.

2.1 Seq2Seq Dialogue Generation

We define the Seq2Seq task-oriented dialogue generation as finding the system response Y according to the input dialogue history X and KB B . Formally, the probability of a response is defined as

$$p(Y | X, B) = \prod_{t=1}^n p(y_t | y_1, \dots, y_{t-1}, X, B), \quad (1)$$

where y_t represents an output token. In a vanilla Seq2Seq task-oriented dialogue system (Eric and Manning, 2017), a long short-term Memory network (LSTM, Hochreiter and Schmidhuber (1997)) is used to encode the dialogue history $X = (x_1, x_2, \dots, x_T)$ (T is the number of tokens in the dialogue history) to produce shared context-sensitive

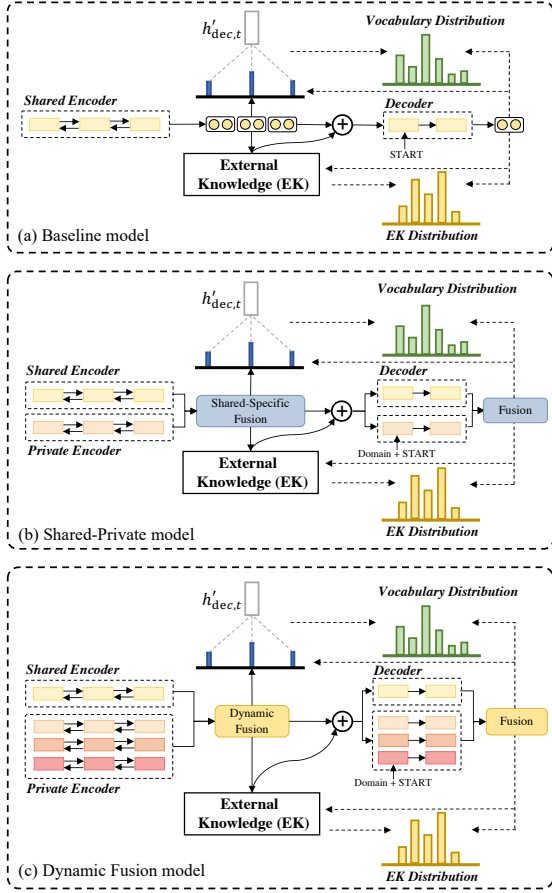


Figure 3: Workflow of our baseline and our proposed model.

hidden states $H = (h_1, h_2, \dots, h_T)$:

$$h_i = \text{BiLSTM}_{\text{enc}} \left(\phi^{\text{emb}}(x_i), h_{i-1} \right), \quad (2)$$

where $\phi^{\text{emb}}(\cdot)$ represents the word embedding matrix. LSTM is also used to repeatedly predict outputs $(y_1, y_2, \dots, y_{t-1})$ by the decoder hidden states $(h_{\text{dec},1}, h_{\text{dec},2}, \dots, h_{\text{dec},t})$. For the generation of y_t , the model first calculates an attentive representation $h'_{\text{dec},t}$ of the dialogue history over the encoding representation H . Then, the concatenation of $h_{\text{dec},t}$ and $h'_{\text{dec},t}$ is projected to the vocabulary space \mathcal{V} by U :

$$o_t = U [h_{\text{dec},t}, h'_{\text{dec},t}], \quad (3)$$

where o_t is the score (logit) for the next token generation. The probability of next token $y_t \in \mathcal{V}$ is finally calculated as:

$$p(y_t | y_1, \dots, y_{t-1}, X, B) = \text{Softmax}(o_t). \quad (4)$$

Different from typical text generation with Seq2seq model, the successful conversations for task-oriented dialogue system heavily depend on accurate knowledge base (KB) queries. We adopt

the global-to-local memory pointer mechanism (GLMP) (Wu et al., 2019a) to query the entities in KB, which has shown the best performance. An external knowledge memory is proposed to store knowledge base (KB) B and dialogue history X . The KB memory is designed for the knowledge source while the dialogue memory is used for directly copying history words. The entities in external knowledge memory are represented in a triple format and stored in the memory module, which can be denoted as $M = [B; X] = (m_1, \dots, m_{b+T})$, where m_i is one of the triplet of M , b and T denotes the number of KB and dialog history respectively. For a k -hop memory network, the external knowledge is composed of a set of trainable embedding matrices $C = (C^1, \dots, C^{k+1})$. We can query knowledge both in encoder and decoder process to enhance model interaction with knowledge module.

Query Knowledge in Encoder We adopt the last hidden state as the initial query vector:

$$q_{\text{enc}}^1 = h_T. \quad (5)$$

In addition, it can loop over k hops and compute the attention weights at each hop k using

$$p_i^k = \text{Softmax}((q_{\text{enc}}^k)^\top c_i^k), \quad (6)$$

where c_i^k is the embedding in i^{th} memory position using the embedding matrix C^k . We obtain the global memory pointer $G = (g_1, \dots, g_{b+T})$ by applying $g_i^k = \text{Sigmoid}((q_{\text{enc}}^k)^\top c_i^k)$, which is used to filter the external knowledge for relevant information for decoding.

Finally, the model reads out the memory o^k by the weighted sum over c^{k+1} and updates the query vector q_{enc}^{k+1} . Formally,

$$o_{\text{enc}}^k = \sum_i p_i^k c_i^{k+1}, \quad q_{\text{enc}}^{k+1} = q_{\text{enc}}^k + o_{\text{enc}}^k. \quad (7)$$

q_{enc}^{k+1} can be seen as the encoded KB information, and is used to initialize the decoder.

Query Knowledge in Decoder we use a sketch tag to denote all the possible slot types that start with a special token. (e.g., @address stands for all the Address). When a sketch tag is generated by Eq. 4 at t timestep, we use the concatenation of the hidden states $h_{\text{dec},t}$ and the attentive representation $h'_{\text{dec},t}$ to query knowledge, which is similar with

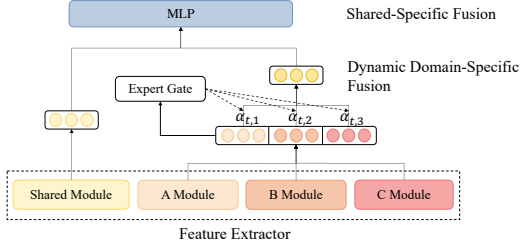


Figure 4: The dynamic fusion layer for fusing domain-shared feature and domain-specific feature.

the process of querying knowledge in the encoder:

$$\mathbf{q}_{\text{dec}}^1 = [\mathbf{h}_{\text{dec},t}, \mathbf{h}'_{\text{dec},t}], \quad (8)$$

$$p_i^k = \text{Softmax}((\mathbf{q}_{\text{dec}}^k)^\top \mathbf{c}_i^k g_i^k). \quad (9)$$

Here, we can treat $P_t = (p_1^k, \dots, p_{b+T}^k)$ as the probabilities of queried knowledge, and select the word with the highest probability from the query result as the generated word.

2.2 Shared-Private Encoder-Decoder Model

The model in section 2.1 is trained over mixed multi-domain datasets and the model parameters are shared across all domains. We call such model as *shared encoder-decoder* model. Here, we propose to use a shared-private framework including a shared encoder-decoder for capturing domain-shared feature and a *private* model for each domain to consider the domain-specific features explicitly. Each instance X goes through both the shared and its corresponding private encoder-decoder.

Enhancing Encoder Given an instance along with its domain, the shared-private encoder-decoder generates a sequence of encoder vectors denoted as $\mathbf{H}_{\text{enc}}^{\{s,d\}}$, including shared and domain-specific representation from corresponding encoder:

$$\begin{aligned} \mathbf{H}_{\text{enc}}^{\{s,d\}} &= (\mathbf{h}_{\text{enc},1}^{\{s,d\}}, \dots, \mathbf{h}_{\text{enc},T}^{\{s,d\}}) \\ &= \text{BiLSTM}_{\text{enc}}^{\{s,d\}}(X). \end{aligned} \quad (10)$$

The final shared-specific encoding representation $\mathbf{H}_{\text{enc}}^f$ is a mixture:

$$\mathbf{H}_{\text{enc}}^f = \mathbf{W}_2(\text{LeakyReLU}(\mathbf{W}_1(\mathbf{H}_{\text{enc}}^s, \mathbf{H}_{\text{enc}}^d))). \quad (11)$$

For ease of exposition, we define the shared-specific fusion function as:

$$\text{shprivate} : (\mathbf{H}_{\text{enc}}^s, \mathbf{H}_{\text{enc}}^d) \rightarrow \mathbf{H}_{\text{enc}}^f. \quad (12)$$

In addition, self-attention has been shown useful for obtaining context information (Zhong et al., 2018). Finally, we follow Zhong et al. (2018) to use self-attention over $\mathbf{H}_{\text{enc}}^f$ to get context vector $\mathbf{c}_{\text{enc}}^f$. We replace \mathbf{h}_T with $\mathbf{c}_{\text{enc}}^f$ in Eq. 5. This makes our query vector combine the domain-shared feature with domain-specific feature.

Enhancing Decoder At t step of the decoder, the private and shared hidden state is:

$$\mathbf{h}_{\text{dec},t}^{\{s,d\}} = \text{LSTM}_{\text{dec},t}^{\{s,d\}}(X). \quad (13)$$

We also apply the shared-specific fusion function to the hidden states and the mixture vector is:

$$\text{shprivate} : (\mathbf{h}_{\text{dec},t}^s, \mathbf{h}_{\text{dec},t}^d) \rightarrow \mathbf{h}_{\text{dec},t}^f. \quad (14)$$

Similarly, we obtain the fused attentive representation $\mathbf{h}_{\text{dec},t}^{f'}$ by applying attention from $\mathbf{h}_{\text{dec},t}^f$ over $\mathbf{H}_{\text{enc}}^f$. Finally, we replace $[\mathbf{h}_{\text{dec},t}, \mathbf{h}'_{\text{dec},t}]$ in Eq. 8 with $[\mathbf{h}_{\text{dec},t}^f, \mathbf{h}_{\text{dec},t}^{f'}]$ which incorporates shared and domain-specific features.

2.3 Dynamic Fusion for Querying Knowledge

The shared-private framework can capture the corresponding specific feature, but neglects the fine-grained relevance across certain subsets of domains. We further propose a dynamic fusion layer to explicitly leverage all domain knowledge, which is shown in Figure 4. Given an instance from any domain, we first put it to multiple private encoder-decoder to obtain domain-specific features from all domains. Next, all domain-specific features are fused by a dynamic domain-specific feature fusion module, followed by a shared-specific feature fusion for obtaining shared-specific features.

Dynamic Domain-Specific Feature Fusion

Given domain-specific features from all domains, a Mixture-of-Experts mechanism (MoE) (Guo et al., 2018) is adapted to dynamically incorporate all domain-specific knowledge for the current input in both encoder and decoder. Now, we give a detailed description on how to fuse the timestep t of decoding and the fusion process is the same to encoder. Given all domain feature representations in t decoding steps: $\{\mathbf{h}_{\text{dec},t}^{d_i}\}_{i=1}^{|\mathbb{D}|}$, where $|\mathbb{D}|$ represents the number of domains, an *expert gate* E takes $\{\mathbf{h}_{\text{dec},t}^{d_i}\}$ as input and outputs a softmax score $\alpha_{t,i}$ that represents the degree of correlation between each domain and the

current input token. We achieve this by a simple feedforward layer:

$$\alpha_t = \text{Softmax}(\mathbf{W} * \mathbf{h}_{\text{dec},t}^d + \mathbf{b}). \quad (15)$$

The final domain-specific feature vector is a mixture of all domain outputs, dictated by the expert gate weights $\alpha_t = (\alpha_{t,1}, \dots, \alpha_{t,|\mathbb{D}|})$, which can be written as $\mathbf{h}_{\text{dec},t}^{d_f} = \sum_i \alpha_{t,i} \mathbf{h}_{\text{dec},t}^{d_i}$.

During training, take the decoder for example, we apply the cross-entropy loss $L_{\text{dec}}^{\text{moe}}$ as the supervision signal for the expert gate to predict the domain of each token in the response, where the expert gate output α_t can be treated as the t^{th} token’s predicted domain probability distribution by multiple private decoder. Hence, the more accurate the domain prediction is, the more correct expert gets:

$$L_{\text{dec}}^{\text{moe}} = - \sum_{t=1}^n \sum_{i=1}^{|\mathbb{D}|} (e_i \cdot \log(\alpha_{t,i} | \theta_s, \theta_{\text{dec}}^m)), \quad (16)$$

where θ_s represents the parameters of encoder-decoder model, θ_{dec}^m represents the parameters of the MoE module (Eq. 15) in the decoder and $e_i \in \{0, 1\}$ represents whether the response with n tokens belongs to the domain d_i . Similarly, we can get the $L_{\text{enc}}^{\text{moe}}$ for the encoder and sum up them as: $\mathcal{L}_{\text{moe}} = L_{\text{enc}}^{\text{moe}} + L_{\text{dec}}^{\text{moe}}$.

\mathcal{L}_{moe} is used to encourage samples from a certain source domain to use the correct expert, and each expert learns corresponding domain-specific features. When a new domain has little or no labeled data, the expert gate can automatically calculate the correlation between different domains with the target domain and thus better transfer knowledge from different source domains in both encoder and decoder module.

Shared-Specific Feature Fusion We directly apply shprivate operation to fuse shared and final domain-specific feature:

$$\text{shprivate} : (\mathbf{h}_{\text{dec},t}^s, \mathbf{h}_{\text{dec},t}^{d_f}) \rightarrow \mathbf{h}_{\text{dec},t}^f. \quad (17)$$

Finally, we denote the dynamic fusion function as $\text{dynamic}(\mathbf{h}_{\text{dec},t}^s, \{\mathbf{h}_{\text{dec},t}^{d_i}\}_{i=1}^{|\mathbb{D}|})$. Similar to Section 2.2, we replace $[\mathbf{h}_{\text{dec},t}, \mathbf{h}_{\text{dec},t}']$ in Eq. 8 with $[\mathbf{h}_{\text{dec},t}^f, \mathbf{h}_{\text{dec},t}^{f'}]$. The other components are kept the same as the *shared-private encoder-decoder* framework.

Dataset	Domains	Train	Dev	Test
SMD	Navigate, Weather, Schedule	2,425	302	304
Multi-WOZ 2.1	Restaurant, Attraction, Hotel	1,839	117	141

Table 1: Statistics of datasets.

Adversarial Training To encourage the model to learn domain-shared features, we apply adversarial learning on the shared encoder and decoder module. Following Liu et al. (2017), a gradient reversal layer (Ganin and Lempitsky, 2014) is introduced after the domain classifier layer. The adversarial training loss is denoted as \mathcal{L}_{adv} . We follow Qin et al. (2019a) and the final loss function of our Dynamic fusion network is defined as:

$$\mathcal{L} = \gamma_b \mathcal{L}_{\text{basic}} + \gamma_m \mathcal{L}_{\text{moe}} + \gamma_a \mathcal{L}_{\text{adv}}, \quad (18)$$

where $\mathcal{L}_{\text{basic}}$ keep the same as GLMP (Wu et al., 2019a), γ_b , γ_m and γ_a are hyper-parameters. More details about $\mathcal{L}_{\text{basic}}$ and \mathcal{L}_{adv} can be found in appendix.

3 Experiments

3.1 Datasets

Two publicly available datasets are used in this paper, which include SMD (Eric et al., 2017) and an extension of Multi-WOZ 2.1 (Budzianowski et al., 2018) that we equip the corresponding KB to every dialogue.¹ The detailed statistics are also presented in Table 1. We follow the same partition as Eric et al. (2017), Madotto et al. (2018) and Wu et al. (2019a) on SMD and (Budzianowski et al., 2018) on Multi-WOZ 2.1.

3.2 Experimental Settings

The dimensionality of the embedding and LSTM hidden units is 128. The dropout ratio we use in our framework is selected from $\{0.1, 0.2\}$ and the batch size from $\{16, 32\}$. In the framework, we adopt the weight typing trick (Wu et al., 2019a). We use Adam (Kingma and Ba, 2015) to optimize the parameters in our model and adopt the suggested hyper-parameters for optimization. All hyper-parameters are selected according to validation set. More details about hyper-parameters can be found in Appendix.

3.3 Baselines

We compare our model with the following state-of-the-art baselines.

¹The constructed datasets will be publicly available for further research.

Model	SMD					Multi-WOZ 2.1				
	BLEU	F1	Navigate F1	Weather F1	Calendar F1	BLEU	F1	Restaurant F1	Attraction F1	Hotel F1
Mem2Seq (Madotto et al., 2018)	12.6	33.4	20.0	32.8	49.3	6.6	21.62	22.4	22.0	21.0
DSR (Wen et al., 2018)	12.7	51.9	52.0	50.4	52.1	9.1	30.0	33.4	28.0	27.1
KB-retriever (Qin et al., 2019b)	13.9	53.7	54.5	52.2	55.6	-	-	-	-	-
GLMP (Wu et al., 2019a)	13.9	60.7	54.6	56.5	72.5	6.9	32.4	38.4	24.4	28.1
Shared-Private framework (Ours)	13.6	61.7	56.3	56.5	72.8	6.6	33.8	39.8	26.0	28.3
Dynamic Fusion framework (Ours)	14.4*	62.7*	57.9*	57.6*	73.1*	9.4*	35.1*	40.9*	28.1*	30.6*

Table 2: Main results. The numbers with * indicate that the improvement of our framework over all baselines is statistically significant with $p < 0.05$ under t-test.

Model	Entity F1 (%)	
	Test	Δ
Full model	62.7	-
w/o Domain-Shared Knowledge Transfer	59.0	3.7
w/o Dynamic Fusion Mechanism	60.9	1.8
w/o Multi-Encoder	61.0	1.7
w/o Multi-Decoder	58.9	3.8
w/o Adversarial Training	61.6	1.1

Table 3: Ablation tests on the SMD test set.

- **Mem2Seq** (Madotto et al., 2018): the model takes dialogue history and KB entities as input and uses a pointer gate to control either generating a vocabulary word or selecting an input as the output.
- **DSR** (Wen et al., 2018): the model leverages dialogue state representation to retrieve the KB implicitly and applies copying mechanism to retrieve entities from knowledge base while decoding.
- **KB-retriever** (Qin et al., 2019b): the model adopts a retriever module to retrieve the most relevant KB row and filter the irrelevant information for the generation process.
- **GLMP** (Wu et al., 2019a): the framework adopts the global-to-local pointer mechanism to query the knowledge base during decoding and achieve state-of-the-art performance.

For *Mem2Seq*, *DSR*, *KB-retriever*², we adopt the reported results from Qin et al. (2019b) and Wu et al. (2019a). For *GLMP*, we rerun their public code to obtain results on same datasets.³

²For Multi-WOZ 2.1 dataset, most dialogs are supported by more than single row, which can not processed by *KB-retriever*, so we compare our framework with it in SMD and Camrest datasets.

³Note that, we find that Wu et al. (2019a) report macro entity F1 as the micro F1, so we rerun their models (<https://github.com/jasonwu0731/GLMP>) and obtain results.

3.4 Results

Follow the prior work (Eric et al., 2017; Madotto et al., 2018; Wen et al., 2018; Wu et al., 2019a; Qin et al., 2019b), we adopt the *BLEU* and *Micro Entity F1* metrics to evaluate model performance. The results on the two datasets are shown in Table 2, we can observe that: 1) The basic *shared-private* framework outperforms the best prior model GLMP in all the datasets. This indicates that the combination of domain-shared and domain-specific features can better enhance each domain performance compared with only utilizing the implicit domain-shared features. 2) Our framework achieves the state-of-the-art performance on two multi-domain task-oriented dialog datasets, namely SMD and Multi-WOZ 2.1. On SMD dataset, our model has the highest BLEU compared with baselines, which shows that our framework can generate more fluent response. More importantly, our model outperforms GLMP by 2.0% overall, 3.3% in the Navigate domain, 1.1% in the Weather domain and 0.6% in Schedule domain on entity F1 metric, which indicates that considering relevance between target domain input and all domains is effective for enhancing performance of each domain. On Multi-Woz 2.1 dataset, the same trend of improvement has been witnessed, which further shows the effectiveness of our framework.

3.5 Analysis

We study the strengths of our model from several perspectives on SMD dataset. We first conduct several ablation experiments to analyze the effect of different components in our framework. Next, we conduct domain adaption experiments to verify the transferability of our framework given a new domain with little or no labeled data. In addition, we provide a visualization of the dynamic fusion layer and case study to better understand how the module affects and contributes to the performance.

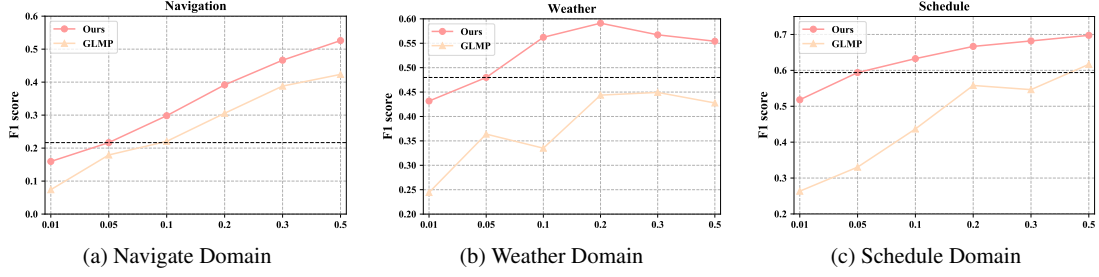


Figure 5: Performance of domain adaption on different subsets of original training data.

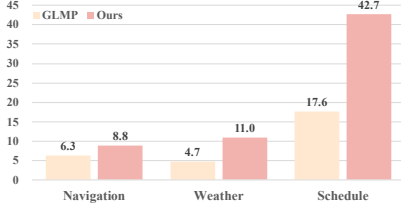


Figure 6: Zero-shot performance (F1 score) on each domain on SMD dataset. The x-axis domain name represents that the domain is unseen and other two domains is the same as original dataset.

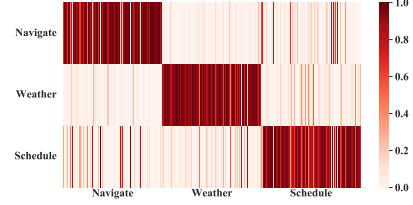


Figure 7: Distribution of Mix-of-the-expert mechanism across source domains for randomly selected 100 examples in each domain on SMD dataset.

Model	Correct	Fluent	Humanlike
GLMP	3.4	3.9	4.0
Our framework	3.6	4.2	4.2
Agreement	53%	61%	74%

Table 4: Human evaluation of responses on the randomly selected dialogue history.

3.5.1 Ablation

Several ablation experiments and results are shown in Table 3. In detail, 1) w/o Domain-shared Knowledge Transfer denotes that we remove domain-shared feature and just keep fused domain-specific feature for generation. 2) w/o Domain Fusion mechanism denotes that we simply sum all domain-specific features rather than use the MOE mechanism to dynamically fuse domain-specific features. 3) w/o Multi-Encoder represents that we remove multi-encoder module and adopt one shared encoder in our framework. 4) w/o Multi-Decoder represents that we remove the multi-decoder module and adopt one shared decoder. 5) w/o Adversarial Training denotes that we remove the adversarial training in experimental setting. Generally, all the proposed components are effective to contribute the final performance. Specifically, we can clearly observe the effectiveness of our dynamic fusion mechanism where w/o domain-specific knowledge fusion causes 1.8% drops and the same trend in removing domain-shared knowledge fusion. This

further verifies that domain-shared and domain-specific feature are benefit for each domain performance.

3.5.2 Domain Adaption

Low-Resource Setting To simulate low-resource setting, we keep two domains unchanged, and the ratio of the except domain from original data varies from [1%, 5%, 10%, 20%, 30%, 50%]. The results are shown in Figure 5. We can find that: (1) Our framework outperforms the GLMP baseline on all ratios of the original dataset. When the data is only 5% of original dataset, our framework outperforms GLMP by 13.9% on all domains averagely. (2) Our framework trained with 5% training dataset can achieve comparable and even better performance compared to GLMP with 50% training dataset on some domains. This implies that our framework effectively transfers knowledge from other domains to achieve better performance for the low-resources new domain.

Zero-Shot Setting Specially, we further evaluate the performance of domain adaption ability on the zero-shot setting given an unseen domain. We randomly remove one domain from the training set, and other domain data remained unchanged to train the model. During test, the unseen domain input use the MoE to automatically calculate the correlation between other domains and the current input and get the results. Results are shown in

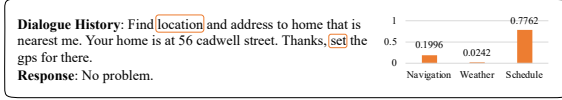


Figure 8: Case of expert gate distribution in SMD dataset. Text segments with red color represents appearing in both schedule and navigation domain.

Figure 6, we can see our model significantly outperforms GLMP on three domains, which further demonstrate the transferability of our framework.

3.5.3 Visualization of Dynamic Fusion Layer

To better understand what our dynamic fusion layer has learned, we visualize the gate distribution for each domain in low-resource (5%) setting, which fuses domain-specific knowledge among various cases. As shown in the Figure 7, for a specific target domain, different examples may have different gate distributions, which indicates that our framework successfully learns how to transfer knowledge between different domains. For example, the navigation column contains 100 examples from its test set and each row show the corresponding expert value. More specifically, in the navigation column, we observe that the expert value in schedule domain is bigger than weather domain, which indicates schedule domain transfers more knowledge to navigation than weather domain.

3.5.4 Case Study

Furthermore, we provide one case for navigation domain and their corresponding expert gate distribution. The cases are generated with 5% training data in the navigation domain and other two domain datasets keep the same, which can better show how the other two domains transfer knowledge to the low-resource domain. As shown in Figure 8, the expert value of schedule domain is bigger than the weather domain, which indicates the schedule contributes more than weather domain. In further exploration, we find word “location” and “set” appear both in navigation and schedule domain, which shows schedule has closer relation with navigation than weather, which indicates our model successfully transfers knowledge from the closest domain.

3.5.5 Human Evaluation

We provide human evaluation on our framework and other baseline models. We randomly generated 100 responses. These responses are based on

distinct dialogue history on the SMD test data. Following Wen et al. (2018) and Qin et al. (2019b), We hired human experts and asked them to judge the quality of the responses according to correctness, fluency, and humanlikeness on a scale from 1 to 5.

Results are illustrated in Table 4. We can see that our framework outperforms GLMP on all metrics, which is consistent with the automatic evaluation.

4 Related Work

Existing end-to-end task-oriented systems can be classified into two main classes. A series of work trains a single model on the mixed multi-domain dataset. Eric et al. (2017) augments the vocabulary distribution by concatenating KB attention to generate entities. Lei et al. (2018) first integrates track dialogue believes in end-to-end task-oriented dialog. Madotto et al. (2018) combines end-to-end memory network (Sukhbaatar et al., 2015) into sequence generation. Gangi Reddy et al. (2019) proposes a multi-level memory architecture which first addresses queries, followed by results and finally each key-value pair within a result. Wu et al. (2019a) proposes a global-to-locally pointer mechanism to query the knowledge base. Compared with their models, our framework can not only explicitly utilize domain-specific knowledge but also consider different relevance between each domain. Another series of work trains a model on each domain separately. Wen et al. (2018) leverages dialogue state representation to retrieve the KB implicitly. Qin et al. (2019b) first adopts the KB-retriever to explicitly query the knowledge base. Their works consider only domain-specific features. In contrast, our framework explicitly leverages domain-shared features across domains.

The shared-private framework has been explored in many other task-oriented dialog components. Liu and Lane (2017) applies a shared-private LSTM to generate shared and domain-specific features. Zhong et al. (2018) proposes a global-local architecture to learn shared feature across all slots and specific feature for each slot. More recently, Zhang et al. (2018) utilizes the shared-private model for text style adaption. In our work, we explore shared-private framework in end-to-end task-oriented dialog to better transfer domain knowledge for querying knowledge base. In addition, we take inspiration from Guo et al. (2018), who successfully apply the mix-of-the-experts (MoE) mechanism in multi-sources domain and cross-lingual

adaption tasks. Our model not only combines the strengths of MoE to incorporate domain-specific feature, but also applies adversarial training to encourage generating shared feature. To our best of knowledge, we are the first to effectively explore shared-private framework in multi-domain end-to-end task oriented dialog.

5 Conclusion

In this paper, we propose to use a shared-private model to investigate explicit modeling domain knowledge for multi-domain dialog. In addition, a dynamic fusion layer is proposed to dynamically capture the correlation between a target domain and all source domains. Experiments on two datasets show the effectiveness of the proposed models. Besides, our model can quickly adapt to a new domain with little annotated data.

6 Acknowledge

We thank Min Xu, Jiapeng Li, Jieru Lin and Zhouyang Li for their insightful discussions. We also thank all anonymous reviewers for their constructive comments. This work was supported by the National Natural Science Foundation of China (NSFC) via grant 61976072, 61632011 and 61772153. Besides, this work also faxed the support via Westlake-BrightDreams Robotics research grant.

References

- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. [MultiWOZ - a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.
- Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. [Key-value retrieval networks for task-oriented dialogue](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.
- Mihail Eric and Christopher Manning. 2017. [A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 468–473, Valencia, Spain. Association for Computational Linguistics.
- Revanth Gangi Reddy, Danish Contractor, Dinesh Raghu, and Sachindra Joshi. 2019. [Multi-level memory for task oriented dialogs](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3744–3754, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.
- Jiang Guo, Darsh Shah, and Regina Barzilay. 2018. [Multi-source domain adaptation with mixture of experts](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4694–4703, Brussels, Belgium. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. [Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.
- Bing Liu and Ian Lane. 2017. Multi-domain adversarial learning for slot filling in spoken language understanding. *arXiv preprint arXiv:1711.11310*.
- Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. [Adversarial multi-task learning for text classification](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada. Association for Computational Linguistics.
- Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. [Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478, Melbourne, Australia. Association for Computational Linguistics.
- Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019a. [A stack-propagation framework with token-level intent detection for spoken](#)

- language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087, Hong Kong, China. Association for Computational Linguistics.
- Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. 2019b. [Entity-consistent end-to-end task-oriented dialogue system with KB retriever](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 133–142, Hong Kong, China. Association for Computational Linguistics.
- Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. [End-to-end memory networks](#). In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448.
- Haoyang Wen, Yijia Liu, Wanxiang Che, Libo Qin, and Ting Liu. 2018. [Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3781–3792, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019a. [Global-to-local memory pointer networks for task-oriented dialogue](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
- Haiming Wu, Yue Zhang, Xi Jin, Yun Xue, and Zhiwen Wang. 2019b. [Shared-private LSTM for multi-domain text classification](#). In *Natural Language Processing and Chinese Computing - 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9-14, 2019, Proceedings, Part II*, pages 116–128.
- Steve J. Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. 2013. [Pomdp-based statistical spoken dialog systems: A review](#). *Proceedings of the IEEE*, 101(5):1160–1179.
- Ye Zhang, Nan Ding, and Radu Soricut. 2018. [SHAPED: Shared-private encoder-decoder for text style adaptation](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1528–1538, New Orleans, Louisiana. Association for Computational Linguistics.
- Victor Zhong, Caiming Xiong, and Richard Socher. 2018. [Global-locally self-attentive encoder for dialogue state tracking](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.

Hyperparameter Name	SMD	Multi-WOZ 2.1
Batch Size	16	32
Hidden Size	128	128
Embedding Size	128	128
Learning Rate	0.001	0.001
Dropout Ratio	0.2	0.1
Teacher Forcing Ratio	0.9	0.9
Number of Memory Network's Hop	3	3

Table 5: Hyperparameters we use for SMD and Multi-WOZ 2.1 dataset.

A Appendices

A.1 Hyperparameters Setting

The hyperparameters used for SMD and Multi-WOZ 2.1 dataset are shown in Table 5.

A.2 Basic Loss Function

The loss \mathcal{L}_{basic} used in our *Shared-Private Encoder-Decoder Model* is the same as GLMP. Different with the standard sequence-to-sequence with attention mechanism model, we use $[\mathbf{h}_{dec,t}^f, \mathbf{h}_{dec,t}^{f'}]$ to replace $[\mathbf{h}_{dec,t}, \mathbf{h}_{dec,t}']$ and then get the sketch word probability distribution P_t^{vocab} . Based on the gold sketch response $\mathbf{Y}^s = (y_1^s, \dots, y_n^s)$, we calculate the standard cross-entropy loss \mathcal{L}_v as follows:

$$P_t^{vocab} = \text{Softmax}(\mathbf{U}[\mathbf{h}_{dec,t}^f, \mathbf{h}_{dec,t}^{f'}]), \quad (19)$$

$$\mathcal{L}_v = \sum_{t=1}^n -\log(P_t^{vocab}(y_t^s)). \quad (20)$$

Given the system response Y , we get the global memory pointer label sequence $G^{label} = (\hat{g}_1, \dots, \hat{g}_{b+T})$ and local memory pointer label sequence $L^{label} = (\hat{l}_1, \dots, \hat{l}_n)$ as follows:

$$\hat{g}_i = \begin{cases} 1 & \text{if Object}(m_i) \in Y \\ 0 & \text{otherwise} \end{cases}, \quad (21)$$

$$\hat{l}_t = \begin{cases} \max(z) & \text{if } \exists z \text{ s.t. } y_t = \text{Object}(m_z) \\ b+T+1 & \text{otherwise} \end{cases}, \quad (22)$$

where m_i represents one triplet in the external knowledge $M = [B; X] = (m_1, \dots, m_{b+T})$ and $\text{Object}(\cdot)$ function is denoted as getting the object word from a triplet.

Then, the \mathcal{L}_g can be written as follows:

$$\mathcal{L}_g = - \sum_{i=1}^{b+T} (\hat{g}_i \cdot \log g_i + (1 - \hat{g}_i) \cdot \log (1 - g_i)). \quad (23)$$

Based on the L^{label} and $P_t = (p_1^k, \dots, p_{b+T}^k)$, we can calculate the standard cross-entropy loss \mathcal{L}_l as follows:

$$\mathcal{L}_l = \sum_{t=1}^n -\log(P_t(\hat{l}_t)). \quad (24)$$

Finally, \mathcal{L}_{basic} is the weighted-sum of three losses:

$$\mathcal{L}_{basic} = \gamma_g \mathcal{L}_g + \gamma_v \mathcal{L}_v + \gamma_l \mathcal{L}_l, \quad (25)$$

where γ_g , γ_v and γ_l are hyperparameters.

A.3 Adversarial Training

We apply a Convolutional Neural Network (CNN) as domain classifier both in the shared encoder and shared decoder to identify the domain of shared representation of dialogue history \mathbf{H}_{enc}^s and response \mathbf{H}_{dec}^s . Take the encoder for example, based on the \mathbf{H}_{enc}^s , we can extract the context representation \mathbf{c}_{enc}^s by CNN and then $\beta_{enc} \in \mathbb{R}^{|\mathbb{D}|}$ can be calculated as follows:

$$\beta_{enc} = \text{Sigmoid}(\text{LeakyReLU}(\mathbf{W}_{enc}(\mathbf{c}_{enc}^s))), \quad (26)$$

Then we train the domain classifier by optimizing its parameters θ_d to minimize the sequence-level binary cross-entropy loss L_{enc}^{adv} as follows:

$$\begin{aligned} \max_{\theta_s} \min_{\theta_d} L_{enc}^{adv} = & - \sum_{i=1}^{|\mathbb{D}|} (e_i \cdot \log(\beta_{enc,i} | \theta_s, \theta_d) \\ & + (1 - e_i) \cdot \log(1 - \beta_{enc,i} | \theta_s, \theta_d)), \end{aligned} \quad (27)$$

where $\beta_{enc,i}$ represents the probability of the input dialogue history belongs to the domain d_i . Similarly, we can get the L_{dec}^{adv} and sum up them as: $\mathcal{L}_{adv} = L_{enc}^{adv} + L_{dec}^{adv}$.

In order to update the encoder-decoder model parameters θ_s underlying the domain classifier, we introduce the gradient reversal layer to reverse the gradient direction which trains our model to extract domain-shared features to confuse the classifier. On the one hand, we train the domain classifier to minimize the domain classification loss. On the other hand, we update the parameters of the network underlying the domain classifier to maximize the domain classification loss, which works adversarially towards the domain classifier. This encourages that our shared encoder and decoder are trained to extract domain-shared features.