Domain-specific Attention with Distributional Signatures for Multi-Domain End-to-end Task-Oriented Dialogue

Xing Ma[1], **Peng Zhang**[1]*, **Feifei Zhao**[2]
[1]*College of Intelligence and Computing, Tianjin University, Tianjin, China*
[2]*Beijing Wenge Technology Co.,Ltd, Beijing, China*
{machine981, pzhang}@tju.edu.cn
feifei.zhao@wengegroup.com

Abstract

The end-to-end task-oriented dialogue system has achieved great success in recent years. Most of these dialogue systems need to accommodate multi-domain dialogue in real-world scenarios. However, due to the high cost of dialogue data annotation and the scarcity of labeled dialogue data, existing methods are difficult to extend to new domains. Therefore, it is essential to use limited data to construct multi-domain dialogue systems. To solve this problem, we propose a novel domain attention module. It uses distributional signatures to construct a multi-domain dialogue system effectively with limited data, which has strong extensibility. We also define an adjacent n-gram pattern to explore potential patterns for dialogue entities. Experimental results show that our approach outperforms the baseline models on most metrics. In the few-shot scenario, we show our method gets a great improvement compared with previous methods while keeping a smaller model scale.

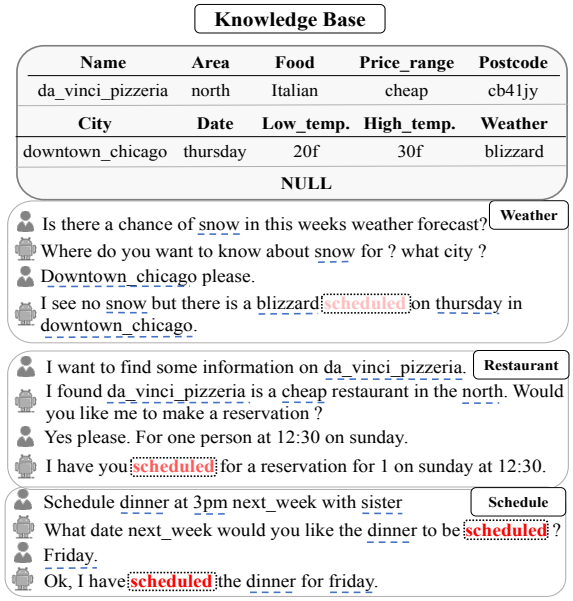


Figure 1: Example of multi-domain dialogue (including weather, restaurant and schedule). Words with blue underlines are entities. The importance of the same word "scheduled" in different domains to dialogue semantics is marked with different levels of red.

1 Introduction

Task-oriented dialogue systems (TOD) aim to assist users in achieving specific goals, such as hotel reservations or weather inquiries, through limited dialogue turns. In contrast with chitchat systems, task-oriented dialogues generate responses based on a specific domain knowledge base (KB). Traditional pipeline methods (Young et al., 2013; Mrkšić et al., 2017) suffer from error propagation and huge cost for intermediate annotations such as dialogue states and actions. Recently, end-to-end methods (Madotto et al., 2018; Wu et al., 2019; Qin et al., 2020; He et al., 2020a; Qin et al., 2021; Ou et al., 2022) have achieved great success by taking the sequence-to-sequence (Seq2Seq) model to generate system responses directly with dialogue history and the specific domain knowledge base. These approaches have the advantages that the dialogue states and actions are latent, which alleviates the need for intermediate annotations.

However, existing end-to-end models are still trained on a large amount of domain-specific dialogue data and the corresponding knowledge base. In practice, task-oriented dialogue systems are often applied to multiple domains. It is difficult for end-to-end models to perform well for domains with limited dialogue data. Hence, it is important to explore how to use the data in the existing domain effectively and transfer the learned knowledge to the new domain with limited data.

Many works are proposed for multiple domains. These methods can be broadly divided into three categories for dealing with different domains in the dataset. The first type of work (Eric and Manning, 2017; Madotto et al., 2018; Wu et al., 2019; He et al., 2020a,b; Raghu et al., 2021; Ou et al., 2022)

*Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 3109–3122

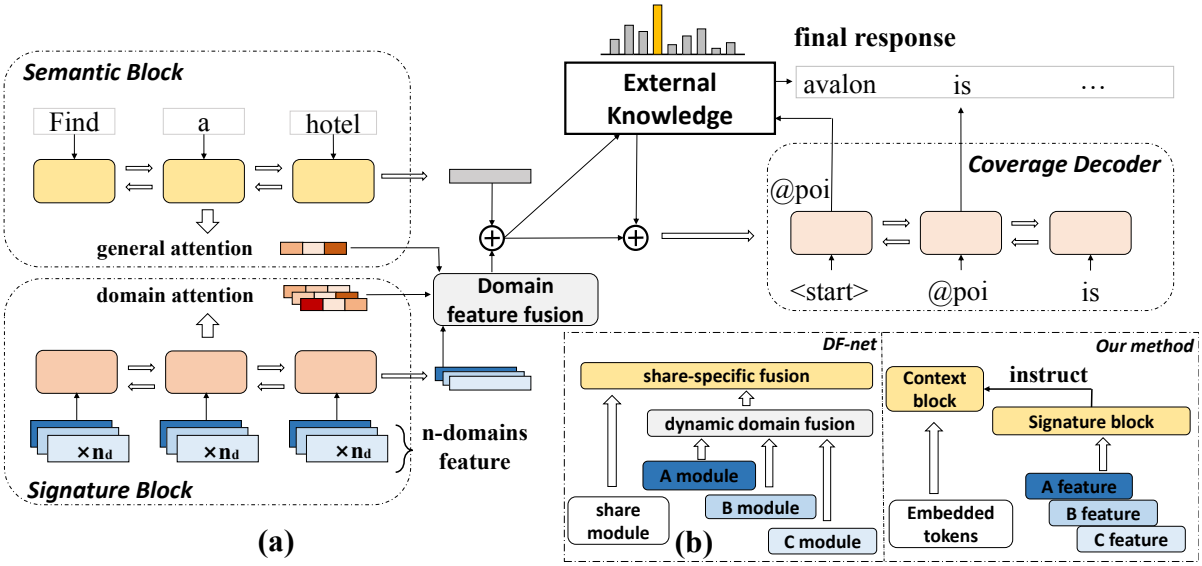


Figure 2: (a) architecture of our model. (b) comparision between our method and DF-net.

does not distinguish between multi-domain data but uses them jointly for training. The second type of work (Wen et al., 2018; Qin et al., 2019) trains separate models for different domain data. The former can make the model learn the shared knowledge of dialogues in various domains and improve the generalization ability of models. However, it can not capture the special knowledge of each domain effectively. The latter can model specific knowledge of dialogues of different domains, but it is challenging to extend to new domains with limited data. The third type of work (Qin et al., 2020) proposes a dynamic fusion mechanism to learn shared and domain-independent knowledge and integrate them with the dynamic fusion mechanism. However, the trained model cannot be flexibly extended to new domains since the number of categories for the domain classifier is predefined. In addition, setting up separate encoders and decoders for each domain adds additional computing overhead.

To address the above issues, we propose a novel domain attention block, which leverages distributional signatures easily extracted from each domain as prior knowledge. As shown in figure 1, we observe that the same word, which appears in different domain dialogue contexts, often has different effects on context understanding and response generation. Furthermore, the KB entities generally have a fixed pattern when appearing in different contexts. We adopt *inverse word frequency*, *domain condition likelihood* to model the former and propose *adjacent n-gram patterns* to model the latter. Instead of one encoder-decoder framework for each domain as figure 2(b), we use a single LSTM to obtain the latent domain knowledge and bridge the gap caused by statistic noise (Bao et al., 2020). A domain feature fusion module is adopted to calculate the similarity between context and each domain feature and fuse the domain-specific attention obtained by the prior knowledge of each domain. We use an auxiliary domain loss to reduce the difference between semantic and signature blocks. Due to learning from the distributional signatures of each domain, our model can better capture general and domain-specific knowledge of multi-domain dialogue.

We conduct experiments on two publicly multi-domain task-oriented dialogue datasets, In-Car assistant(Eric et al., 2017) and Multi-WOZ 2.1(Budzianowski et al., 2018). Our model outperforms baseline models on most metrics. In a low resource setting, our model outperforms the prior state-of-the-art model by 1.4% in entity F1 and by 1.8% in BLEU on In-Car Assistant dataset.

2 Methodology

As shown in figure 2(a), given dialogue of domain d ($d \in \mathcal{D}$, $\mathcal{D}$ is the set of all domains) between user and system, our model takes the tokens $X = (x_1, x_2, ..., x_T)$ from dialogue history and the corresponding multi-domain distributional signatures $S_d = (s_{1,d}, s_{2,d}, ..., s_{T,d})$ into semantic and signature block respectively. Then we use the context vector obtained by two blocks to initialize the knowledge module. Finally, the decoder generates final responses sequentially with the KB read-out

vector and context hidden state.

2.1 Distributional Signatures

We obtain prior distributional signatures from multi-domain dialogue data to learn the general and domain-specific knowledge better.

Adjacent N-gram Patterns We propose adjacent n-gram patterns to model the fixed patterns of dialogue data. It is calculated through the conditional probability $\boldsymbol{p_{cond}}$ of a forward or backward adjacent n-grams $\hat{x}_n$ in context.

$$\begin{cases} \hat{x}_n = (x_{i+1*I_{sign}}, x_{i+2*I_{sign}}, x_{i+n*I_{sign}}) \\ \boldsymbol{p}_d^n(\hat{x}_n) = \dfrac{1}{v}\displaystyle\sum_{x_j}^{v} \dfrac{\varepsilon}{\varepsilon + P_d^n(x_j \mid \hat{x}_n)} \end{cases} \tag{1}$$

where v is the vocab size. I_{sign} is 1 for forward n-gram -1 for backward n-gram.

Variable words near a fixed pattern are often related to entities. In other words, x_i with larger $\boldsymbol{p}_d^n$ adjacent n-grams pattern is more likely to be an entity in the dialogue domain as shown in figure 3. We take both forward and backward adjacent n-grams $\boldsymbol{p}_d^n(\hat{x}_n)$ as a feature of word x_i. For implementation, we use *nltk toolkit* [1] (Bird et al., 2009) to calculate the n-gram frequency of all dialog contexts.

Inverse Word Frequency Word frequency is an important measure of the information that a word provides in a dialog. Following Bao et al. (2020), we reduce the weight of high-frequency words and increase the weight of low-frequency words. We define domain inverse word frequency $\boldsymbol{iwf_d}$.

$$\boldsymbol{iwf_d}(x) = \frac{\varepsilon}{\varepsilon + P_d(x)}, \tag{2}$$

where $\varepsilon = 10^{-5}$, x is the word of domain d, and $P_d(x)$ is the unigram likelihood over domain d data. The general inverse word frequency $\boldsymbol{iwf_g}$ is calculated in a similar way, in which $P_d(x)$ is replaced by $P_g(x)$. $P_g(x)$ is the unigram likelihood over the whole dataset.

Domain Conditional Likelihood Important words in a domain often play an essential role in the semantics of the dialogue in that domain. Therefore, we define a *domain conditional likelihood* $\boldsymbol{c_g}$ to estimate the role of a word in some domain.

$$\boldsymbol{c_d}(x) = P(d \mid x) \tag{3}$$

$$\boldsymbol{c_g}(x) = \frac{\varepsilon}{\varepsilon + \mathcal{H}(\boldsymbol{c_d}(x))} \tag{4}$$

where $\boldsymbol{c_d}(x)$ is determined by conditional probability instead of predicting by using a regularized linear classifier (Bao et al., 2020). We employ an entropy operator $\mathcal{H}$ to measure the uncertainty of domain d.

In practice, we set the zero signatures mentioned above to ε. For adjacent n-gram patterns, we set zero $P_d^n(x_j \mid \hat{x}_n)$ to ε to calculate the final adjacent n-gram patterns $\boldsymbol{p}_d^n(\hat{x}_n)$. Finally, we use the concatenation of all signatures as the domain-specific feature of a word.

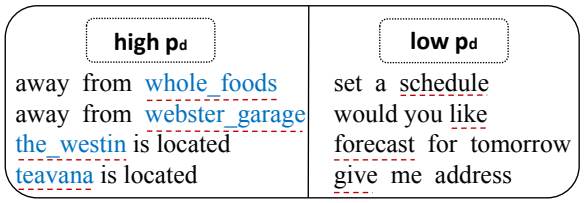


Figure 3: Example of words with low and high $\boldsymbol{p_d}$ adjacent bi-grams. Words with red underlines is x_i. The entities are marked blue.

2.2 Context Encoder

We divide the context encoder into two blocks to encode the semantics and distributional signatures of the contexts, respectively.

Semantic Block We first embed the dialogue history tokens $X = (x_1, x_2, ..., x_T)$ into a fixed-dimensional word vector by using an embedding matrix. Following Gangi Reddy et al. (2019), we use the GloVe word vector to initialize the embedding weight. Then, we employ a bidirectional GRU (Cho et al., 2014) to encode the embedded dialogue history.

$$\boldsymbol{h}_i^{se} = \text{BiGRU}(\phi^{emb}(x_i), \boldsymbol{h}_{i-1}^{se}) \tag{5}$$

where $\phi^{emb}(\cdot)$ is the word embedding matrix. All context hidden states $\boldsymbol{H^{se}} = (\boldsymbol{h}_1^{se}, \boldsymbol{h}_2^{se}, ..., \boldsymbol{h}_T^{se})$ are obtained by this way. Following Zhong et al. (2018), we adopt simple self-attention over $\boldsymbol{H_{ctx}}$ to get the semantic attention of context.

$$\boldsymbol{u}_i = W_{se,2}(\sigma(W_{se,1}\boldsymbol{h}_i^{se} + b_{se,1})) \tag{6}$$

$$\boldsymbol{a}_i = \frac{\exp(\boldsymbol{u}_i)}{\sum_j \exp(\boldsymbol{u}_j)} \tag{7}$$

[1] https://github.com/nltk/nltk

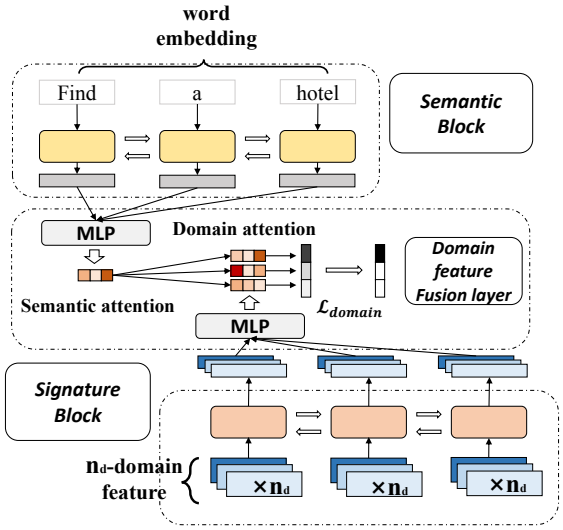


Figure 4: Detailed structure of encoder.

where σ is an activation function. $W_{se,1}$, $W_{se,2}$, $b_{se,1}$ are trainable parameter. Finally, we get semantic attention $\boldsymbol{Att^{se}} = (\boldsymbol{a}_1^{se}, \boldsymbol{a}_2^{se}, ..., \boldsymbol{a}_n^{se})$ of the dialogue history.

Signature Block We leverage the distributional signatures $\boldsymbol{s_i}$ to capture the general and domain-specific knowledge. However, there is much noise in these data, which may interfere with the training process. We take a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to bridge the gap following Bao et al. (2020).

$$\boldsymbol{s}_{i,d} = [\boldsymbol{iwf}_d(x_i), \boldsymbol{iwf}_g(x_i), \boldsymbol{c}_d(x_i), \boldsymbol{c}_g(x_i), \boldsymbol{p}_d^n(x_i)] \tag{8}$$

$$\boldsymbol{h}_{i,d}^{si} = \text{BiLSTM}(\boldsymbol{s}_i, \boldsymbol{h}_{i-1}^{si}) \tag{9}$$

where $\boldsymbol{p}_d^n(x_i)$ is concatenation of $[\boldsymbol{p}_d^{n_p}(x_i)], n_p \in \{2, 3, 4\}$. Similar to the semantic block, we adopt a self-attention layer over signatures hidden states $\boldsymbol{H_d^{si}} = (\boldsymbol{h}_{1,d}^{si}, \boldsymbol{h}_{2,d}^{si}, ..., \boldsymbol{h}_{T,d}^{si})$ to get signature attention $\boldsymbol{Att_d^{si}} = (\boldsymbol{a}_{1,d}^{si}, \boldsymbol{a}_{2,d}^{si}, ..., \boldsymbol{a}_{T,d}^{si})$. Finally, we obtain n_d signature attention and single context attention.

Domain Feature Fusion To fuse multiple domain features, we propose a domain feature fusion function based similarity between semantic attention $\boldsymbol{Att^{se}}$ and signature attention $\boldsymbol{Att_d^{si}}$ as shown in figure 4.

$$\boldsymbol{v}_d = \langle \boldsymbol{Att^{se}}, \boldsymbol{Att_d^{si}} \rangle \tag{10}$$

$$\boldsymbol{a}_d = \frac{\exp(\boldsymbol{v}_d)}{\sum_{d_i} \exp(\boldsymbol{v}_{d_i})} \tag{11}$$

where $\langle \cdot \rangle$ is scalar product function. We define $\boldsymbol{P_{domain}} = [a_1, a_2, ...a_{d_n}]$. Then, We get the domain weights $\boldsymbol{a}_d$ to merge all domain-specific attention.

$$\boldsymbol{Att^{si}} = \sum_d \boldsymbol{a}_d \boldsymbol{Att_d^{si}} \tag{12}$$

Finally, we use the fused domain attention $\boldsymbol{Att_d^{si}}$ and semantic attention $\boldsymbol{Att^{se}}$ over $\boldsymbol{H^{se}}$ to get the context vector $\boldsymbol{c}$

$$\boldsymbol{c_{enc}} = W_{enc}[\sum_i \boldsymbol{a}_i^{se} \boldsymbol{h}_i^{se}, \sum_i \boldsymbol{a}_i^{si} \boldsymbol{h}_i^{se}] \tag{13}$$

2.3 Knowledge Module

To obtain the knowledge needed to generate system responses, the model needs to interact with the knowledge base and get query results. We adopt the memory network with the global-to-local pointer mechanism (Wu et al., 2019) to encode and query the external knowledge.

The external knowledge includes the corresponding knowledge base $\mathcal{K}_d$ and dialogue history X. The ith entity triplet $e_i = (e_{i,sub}, e_{i,rel}, e_{i,obj})$ is represented as $\boldsymbol{c}_i^m = \text{BOW}(C^m(e))$. $\text{BOW}(\cdot)$ is a bag of word function and C^m is an embedding matrix for a k-hop memory network, where $m \in \{1, 2, ..., k\}$. We initialize the memory representation in the encoder stage and query the memory model sequentially in the decoder stage.

Initialize Memory Representation We use the final context vector $\boldsymbol{c_{enc}}$ as the initial vector $\boldsymbol{q}_{init}^1$ to initialize the memory module. Then, we get the global pointer $\boldsymbol{g}_i^m$ through the k-hop mechanism. The whole initialize process is calculated as follows:

$$\boldsymbol{p}_i^m = Softmax(\langle \boldsymbol{q}_{init}^m, \boldsymbol{c}_i^m \rangle) \tag{14}$$

$$\boldsymbol{g}_i^m = Sigmoid(\langle \boldsymbol{q}_{init}^m, \boldsymbol{c}_i^m \rangle) \tag{15}$$

$$\boldsymbol{o}_{init}^m = \sum_i \boldsymbol{p}_i^m \boldsymbol{c}_i^{m+1} \tag{16}$$

$$\boldsymbol{q}_{init}^{m+1} = \boldsymbol{q}_{init}^m + \boldsymbol{o}_{init}^m \tag{17}$$

where $\boldsymbol{o}_{init}^m$ is the weighted sum over $\boldsymbol{c}_i^m$. We obtain a memory read-out vector $\boldsymbol{q}_{init}^{k+1}$ to initialize the decoder. The last hop global pointer $\boldsymbol{g}_i^k$ is used to strengthen the KB entities representation that appears in contexts in the decoder stage.

Query Memory Module We get a context vector $\boldsymbol{c}_{t,dec}$ in the t step of the decoder stage and use it to query the memory module. We apply the

global pointer to give different weights to each entity. Then, we calculate the similarity between context vector and entity representations based on the dot product to obtain the copy probability of entities.

$$\boldsymbol{p}_{i,t}^{m} = Softmax(\langle \boldsymbol{c}_{t,dec}, \boldsymbol{c}_i^m \boldsymbol{g}_i^k \rangle) \quad (18)$$

We define the query result as the last hop probability $\boldsymbol{P}_t^{kb} = [\boldsymbol{p}_{1,t}^k, \boldsymbol{p}_{2,t}^k, ..., \boldsymbol{p}_{T+b,t}^k,]$. We can select the word for generated responses with the highest $\boldsymbol{p}_{i,t}^k$.

2.4 Attention Decoder

We apply a sketch decoder to first generate a coarse response in which sketch tags substitute all the entities. For example, a sentence *"dish_parking is five_miles away"* is written as *"@poi is @distance away"*. Then we use the copied entity as mentioned in section 2.3 to replace the sketch tag.

We adopt a Bi-GRU to generate coarse responses and use the concatenation of KB read-out vector $\boldsymbol{q}_{init}^{k+1}$ and the last context hidden states $\boldsymbol{h}_T^{\boldsymbol{se}}$ as the initial vector (different with memory initialization in section 2.3)

$$\boldsymbol{h}_0^{dec} = W_{concat}[\boldsymbol{q}_{init}^{k+1}, \boldsymbol{h}_T^{\boldsymbol{se}}] + b_{concat} \quad (19)$$

$$\boldsymbol{h}_{t+1}^{dec} = \text{BiGRU}(\phi^{emb}(\boldsymbol{x}_t), \boldsymbol{h}_t^{dec}) \quad (20)$$

where $\boldsymbol{x}_t$ is the generated token at t timestep of the decoder. We adopt the attention mechanism (Bahdanau et al., 2015) to reduce the information loss between the encoder and decoder. In addition, we add the coverage mechanism (See et al., 2017) to reduce excessive attention on specific contexts.

$$\boldsymbol{e}_i^t = v^T tanh(W_e \boldsymbol{h}_i^{se}, W_d \boldsymbol{h}_t^{dec}, w_c \boldsymbol{c}_i^t + b_a) \quad (21)$$

$$\boldsymbol{a}_t = Softmax(\boldsymbol{e}^t) \quad (22)$$

$$\boldsymbol{c}_{dec,t} = \sum_i \boldsymbol{a}_i^t \boldsymbol{h}_t^{dec} \quad (23)$$

$$\boldsymbol{P}_t^{vocab} = Softmax(V[\boldsymbol{h}_t^{dec}, \boldsymbol{c}_{dec,t}] + b) \quad (24)$$

where $\boldsymbol{c}_{dec,t}$ is used as a query vector to interact with the knowledge module.

Finally, we generate the coarse responses through the final probability $\boldsymbol{P}_t^{vocab}$. If a sketch tag is in coarse responses, we use $\boldsymbol{P}_t^{kb}$ to obtain the corresponding entity.

2.5 Joint Training

To encourage the semantic module to learn more from distributional signatures modules, we design a domain feature loss $\mathcal{L}_{domain}$ to close the gap between the two blocks. The final loss function is defined as:

$$\mathcal{L}_{domain} = \sum_d -y_d \log a_d \quad (25)$$

$$\mathcal{L}_{coverage} = \sum_i min(a_i^t, c_i^t) \quad (26)$$

$$\mathcal{L} = \mathcal{L}_{basic} + \mathcal{L}_{domain} + \mathcal{L}_{coverage} \quad (27)$$

where $y_d \in \{0, 1\}$ and $\mathcal{L}_{basic}$ is same as GLMP (Wu et al., 2019). The details about $\mathcal{L}_{basic}$ can be found in appendix A.1.

3 Experiment

3.1 Datasets

We conducted the experiments on two publicly available task-oriented dialogue datasets, which include two multi-domain datasets: In-Car Assistant (Eric et al., 2017) and Multi-WOZ 2.1 (Budzianowski et al., 2018). We follow the partition as Madotto et al. (2018); Wu et al. (2019) on In-Car Assistant and Qin et al. (2020) on Multi-WOZ 2.1. More details about the two datasets are presented in appendix A.2.

3.2 Experimental Settings

We set n to $\{2, 3, 4\}$ for adjacent n-gram pattern signatures. The model is trained using Adam optimizer (Kingma and Ba, 2015) and learning rate starts from $1e^{-3}$ to $1e^{-4}$. We select dropout rate from $\{0.2, 0.3\}$ and batch size from $\{8, 16, 32\}$. We also use the pre-trained GloVe vector (Pennington et al., 2014) to initialize our embedding. The words not in GloVe are initialized using Glorot uniform distribution (Glorot and Bengio, 2010). The hidden units of GRU are set to the same dimension with embedding. We adopt an exponential schedule sampling(Bengio et al., 2015) in the decoder stage. You can find more details about hyper-parameters in appendix A.3.

3.3 Baselines

(1) **Mem2Seq** (Madotto et al., 2018) adopts a memory network to encode KB and combines the vocabulary and entity probability through a hard gate.
(2) **GLMP** (Wu et al., 2019) applies the global-to-local pointer mechanism to query the knowledge module.
(3) **KB-retriever** (Qin et al., 2019) retrieves the most relevant KB row and filters the irrelevant information in the whole process.

Model	In-Car Assistant					Multi-WOZ 2.1				
	BLEU	F1	Calendar F1	Weather F1	Navigate F1	BLEU	F1	Restaurant F1	Attraction F1	Hotel F1
Mem2Seq	12.6	33.4	49.3	32.8	20.0	6.6	21.6	22.4	22.0	21.0
GLMP	13.9	60.7	54.6	56.5	53.0	6.9	32.4	38.4	24.4	28.1
KB-retriever	17.2	59.0	71.8	57.8	52.5	-	-	-	-	-
Fg2Seq	16.8	61.1	73.3	57.4	56.1	**13.5**	36.0	40.4	41.7	30.9
DA-HIMN	16.2	61.2	73.8	60.6	54.3	9.2	37.7	39.3	37.4	36.1
DFNet	14.4	62.7	73.1	57.9	57.6	9.4	35.1	40.9	28.1	30.6
CD-NET	17.8	62.9	**75.4**	**61.3**	56.7	11.9	38.7	**41.7**	38.9	36.3
Our model	**18.0**	**63.0**	72.3	55.2	**61.4**	12.3	**39.5**	41.2	**45.5**	**37.0**

Table 1: Performance of our model and baselines on the In-Car Assistant and Multi-WOZ 2.1 datasets. The best results are bolded, and the second best results are underlined.

Model	Entity F1	
	test	Δ
full model	63.0	
w/o Domain Loss	62.0	1.0
+ w/o Signature Block	61.4	1.6
w/o coverage attention	61.8	1.2
origin model	60.7	2.3

Table 2: Ablation study on In-Car Assistant dataset.

(4) **DFnet** (Qin et al., 2020) adopts a dynamic fusion mechanism to learn shared knowledge and domain-independent knowledge.
(5) **Fg2Seq** (He et al., 2020a) uses a flow operation to strengthen the connection between the dialogue history and the knowledge base.
(6) **CD-NET** (Raghu et al., 2021) proposes a pairwise similarity-based KB distillation to enhance the relation between KB and context.
(7) **DA-HIMN** (Ou et al., 2022) combines request-aware with KB-aware to better capture the latest request of users.

We run their code for *DA-HIMN* to obtain the results on Multi-WOZ2.1. For the rest of baselines, we adopt the results reported from their paper.

3.4 Results

We adopt the micro Entity F1 and BLEU as our evaluation metrics following (Madotto et al., 2018; Wu et al., 2019; Qin et al., 2020). The results on the two datasets are shown in Tabel 1. We can see that our model outperforms baselines on most metrics. We mainly compare our model with *GLMP* and *DF-net*, which are similar frameworks. Our model outperforms *DF-net* 0.3% and 4.4% in entity F1 on In-Car Assistant and Multi-WOZ2.1, respectively. We also exceed 3.3% over *DF-net* in BLEU on average. In addition, our model also outperforms *GLMP* 4.7% and 4.8% in entity F1 and BLEU on average. The results indicate that the signature block and all distributional signatures effectively help the model to learn different domain knowledge and mitigate the domain bias.

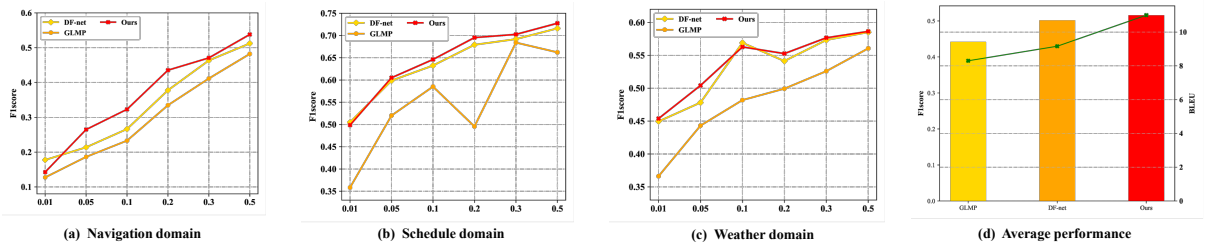
3.5 Analysis

We discuss the validity of the model through experiments on In-Car Assistant dataset from the following aspects. We first conduct ablation experiments to verify the effectiveness of our model and explore the role of different signatures. Then we evaluate our model in a low resource setting and calculate the model size compared with *DF-net* and *GLMP*. Finally, we use practical cases to demonstrate the effectiveness of the method.

3.5.1 Ablation

Ablation of Components We conduct some ablation experiments on our model. The results are shown in Table 2. (1) We first remove the domain loss and just keep the signature block. Our model achieves 62.0% in entity F1 with a drop of 1.0%. The performance drop demonstrates that domain loss is critical for instructing the semantic module. (2) Based on (1), we remove the whole signature block, and F1 score drops to 61.4%, indicating that distribution signatures obviously contribute to the model's performance. (3) Then, we remove the coverage context attention mechanism. The performance of the model decreases significantly. The covering attention mechanism indirectly affects the performance of querying knowledge base in the generation process by influencing the generation process of the model.

Ablation of signatures We evaluate our model with different signatures to explore their role in the model. We mainly care about the relation between word features (*inverse word frequency* and *domain conditional likelihood*) and *adjacent n-gram patterns*. In addition, we also study the n value of adjacent n-grams. The experiment results are shown



(a) Navigation domain (b) Schedule domain (c) Weather domain (d) Average performance

Figure 5: Main results of domain adaption on In-Car Asisstant dataset.

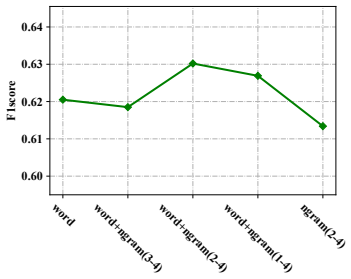


Figure 6: Ablation of different signatures combinations.

Model	Model Scale	Size growth per domain
GLMP	8.68	3.8 (0)
DF-net	34.07	8.4 (4.6)
Our model	12.23	3.8 (0)

Table 3: Comparison of model size and size growth with the number of domains (MB). Model size growth without word embeddings is in parentheses.

in figure 6. It can be seen that model only using word features gets a good result because the word features have a strong relationship with the domain. Our model achieves the best performance when $n \in \{2, 3, 4\}$. When n set adds 1, it has a little drop in entity F1. This may be caused by many short patterns unrelated to entities interfering with the learning process of the model. The performance drops significantly when n set removes 4, which indicates our model suffers from the lack of short pattern features. We also observe that model only using adjacent n-gram patterns has bad performance. The domain loss can not capture the relation of different domains without word features.

3.5.2 Domain Adaption

We follow Qin et al. (2020) to conduct domain adaption experiments. We keep two domain data unchanged and use different ratio resources of the last domain data. The ratio is selected from [1%,5%,10%,20%,30%,50%]. We adopt the same GloVe vector and dimension to *DF-net* and *GLMP* to remove the influence of irrelevant factors. As shown in figure 5, We can observe that our model achieves competitive results with *DF-net* and has significant improvement over *GLMP* in total. Particularly, our model gets 1.8% higher in BLEU than *DF-net*. It is because we use the hidden states instead of context vector over attention to initialize the decoder. In addition, we find that our model performs poorly when the training set ratio is low. It indicates that inaccurate distributional signatures of low-ratio training set bring bias to our model.

3.5.3 Model Scale

We compare our model size with other baselines in the same setting as shown in Table 3. Our model has 3.6MB larger than *GLMP* and 21.8MB smaller than *DF-net*. In addition, *DF-net* sets up an independent encoder and decoder for each domain. The model size grows by 4.6MB per extra domain, except for word embedding. However, we achieve competitive results with a smaller model scale, which does not change much with the increase of the dialogue domains. It denotes that our signatures block improves the efficiency of our model.

3.5.4 Case Study

We use some cases of dataset to analyze the effects of our method, which are given in Table 4. We observe that our model generates more fluent responses with right information. In case 1, our model generates right entities compared with *DF-net* and the response is more fluent than *GLMP*. The question of case 2 mentions *'next'*, which requires the model to make judgments based on the time attribute of dinners. Our model outperforms other baselines and obtains the right information from KB in this situation. In addition, our model can better generate complex responses like case 3 for which other baseline models fail to produce smooth sentences. Our generated response is informative and grammatically correct.

	dialog examples
KB	[palo_alto_garage_r, 481_amaranta_ave, parking_garage, car_collision_nearby, 1_miles]
Question	where is a nearby parking_garage ?
Response	the nearest one is palo_alto_garage_r , it s just 1_miles away
GLMP	[**palo_alto_garage_r**] is [**1_miles**] away. (*Right info.*)
DF-net	[**stanford_oval_parking**] is [**1_miles**] away. (*Wrong poi info.*)
Our model	the closest parking_garage is [**palo_alto_garage_r**] , it s [**1_miles away**]. (*Right info. and fluent response*)
KB	[dinner, jon, the_11th, 6pm]
Question	what time is my next dinner event and who will be attending ?
Response	the one at 6pm with jon is on the_11th.
GLMP	your next dinner is on the_11th at [**5pm**] with [**jon**]. (*Wrong time info.*)
DF-net	you have two dinner scheduled one on the_20th with [jon] and one on the_11th with tom and one on [**6pm**] with [**jon**]. (*Wrong problem understanding*)
Our model	your next dinner is on the_11th at [**6pm**] with [**jon**]. (*Right understanding and info.*)
KB	[manhattan, wednesday, high, 50f, low, 40f, hail]
Question	what will be the temperature in manhattan on wednesday ?
Response	it appears as if there will be a low of 40f and a high of 50f.
GLMP	the temperature in manhattan will be on wednesday. (*Wrong senmantics*)
DF-net	the weather in manhattan will be wednesday on monday. (*Wrong senmantics*)
Our model	the temperature in manhattan will be low of [**40f**] , high of [**50f**] on wednesday. (*Right temp. info.*)

Table 4: Comparing the generated responses with *GLMP* and *DF-net* on examples of different domain.

4 Related Work

Existing end-to-end approaches to modeling multi-domain datasets can be divided into three categories. The first strand of work trains the model on the mixed data directly. Madotto et al. (2018) first adopts end-to-end memory network (Sukhbaatar et al., 2015) to encode KB items and dialogue contexts. Wu et al. (2019) proposes a global-to-local pointer mechanism to improve the accuracy of querying KB based on the memory network. Our model retains the main framework of Wu et al. (2019). He et al. (2020a) uses a flow operation to strengthen the connection between the dialogue history and the knowledge base. Raghu et al. (2021) also propose a pairwise similarity-based KB distillation to achieve the same purpose as He et al. (2020a). Ou et al. (2022) combines request-aware with KB-aware to better capture the latest request of users. Xie et al. (2022) models task-oriented dialogues as a text-to-text task and fine-tunes the T5 model (Raffel et al., 2020) on the mixed dataset. These works treat data from different domains in the same way, which ignores domain-specific knowledge. The second strand of work trains separate models for each domain. Wen et al. (2018) use the dialog state representation of some domain to query the knowledge base. Qin et al. (2019) restricts the query result from a single KB record. They both only focus on domain-specific knowledge and lack general knowledge. The third strand of work (Qin et al., 2020) proposes a dynamic fusion network to handle multi-domain dialog, which needs multiple encoder-decoder for each domain and lacks flexibility.

The distributional signature of text data contains rich semantic and structural knowledge. Bao et al. (2020) uses the distributional signature to generate general and class-specific attention and improve text classification performance. In our work, we leverage the dialogue data signature of different domains instead of classes. Following Bao et al. (2020), we employ a Bi-LSTM (Hochreiter and Schmidhuber, 1997) to bridge the gap caused by statistic noise. In addition, we take inspiration from Zhong et al. (2010), which proposes an effective pattern taxonomy model. We design adjacent n-gram patterns to discover entities better in the dialogue context. To our best knowledge, we are the first to use distributional signatures to model multi-domain task-oriented dialog.

5 Conclusion

In this work, we propose a domain attention module with distributional signatures of dialogue corpus to capture domain-specific knowledge. We combine the features of different domains in an extensible way, and a domain loss is used to instruct our model to learn better from signatures. In addition, we define a *adjacent n-gram pattern* to mine the KB entities in the dialogue context. We also adopt attention with a coverage mechanism to improve the quality of generated responses. Extensive experiments have demonstrated the effectiveness of our method.

6 Acknowledgment

This work is supported in part by the Natural Science Foundation of China (grant No.62276188 and No.61876129), TJU-Wenge joint laboratory funding and MindSpore.

7 Limitation

Although our model achieves competitive results with baseline models, some limitations are summarized as follows.

1. The process of extracting data distributional signatures is time-consuming, especially for datasets with more diverse dialogue patterns. The process of calculating adjacent n-grams is slow. In addition, repeated string manipulation for long texts also needs to be optimized

2. The experiment results are easily affected by the fluctuation of hyper-parameters, especially the signature block hidden size. There is some noise in the distributional signatures. Under different hyper-parameters, noise may have different effects and directly affect experiment results.

3. Our model performs poorly when the training set is too small. The distributional signatures of small data interfere with the model.

8 Ethics Statement

This paper proposes a domain attention module with distributional signatures to better learn the domain-specific and general knowledge. We also define an adjacent n-gram pattern to mine the entities in the context. We work within the purview of acceptable privacy practices and strictly follow the data usage policy. We use public datasets and consist of their intended use in experiments. We described our experimental setting in detail to ensure reproducibility. We neither introduce any social/ethical bias to the model nor amplify any bias in the data, and our work will not have any social consequences or ethical issues.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural 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 1171–1179.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

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.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. 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.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international*

conference on artificial intelligence and statistics, pages 249–256. JMLR Workshop and Conference Proceedings.

Zhenhao He, Yuhong He, Qingyao Wu, and Jian Chen. 2020a. Fg2seq: Effectively encoding knowledge for end-to-end task-oriented dialog. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 8029–8033. IEEE.

Zhenhao He, Jiachun Wang, and Jian Chen. 2020b. Task-oriented dialog generation with enhanced entity representation. In *INTERSPEECH*, pages 3905–3909.

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*.

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.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Yangyang Ou, Peng Zhang, Jing Zhang, Hui Gao, and Xing Ma. 2022. Incorporating dual-aware with hierarchical interactive memory networks for task-oriented dialogue.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Bowen Qin, Min Yang, Lidong Bing, Qingshan Jiang, Chengming Li, and Ruifeng Xu. 2021. Exploring auxiliary reasoning tasks for task-oriented dialog systems with meta cooperative learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13701–13708.

Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. 2019. 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.

Libo Qin, Xiao Xu, Wanxiang Che, Yue Zhang, and Ting Liu. 2020. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6344–6354, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Dinesh Raghu, Atishya Jain, Mausam, and Sachindra Joshi. 2021. Constraint based knowledge base distillation in end-to-end task oriented dialogs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5051–5061, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. 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. 2019. 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*. OpenReview.net.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *ArXiv preprint*, abs/2201.05966.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Ning Zhong, Yuefeng Li, and Sheng-Tang Wu. 2010. Effective pattern discovery for text mining. *IEEE transactions on knowledge and data engineering*, 24(1):30–44.

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.

A Appendix

A.1 Loss function

The terms $\mathcal{L}_{basic}$ in loss function is same as Wu et al. (2019).

$$\mathcal{L}_{basic} = \mathcal{L}_g + \mathcal{L}_v + \mathcal{L}_l \tag{28}$$

where $\mathcal{L}_v$ and $\mathcal{L}_g$ are the cross entropy of token and local pointer, $\mathcal{L}_l$ is the binary cross entropy of global pointer. The labels of global and local pointer are determined by entities of responses.

$$\hat{g}_m = \begin{cases} 0 & if\ Object(e_m) \in Y \\ 1 & otherwise \end{cases} \tag{29}$$

$$\hat{l}_t = \begin{cases} max(z) & if \exists z,\ s.t.\ y_t = Object(e_z) \\ T + b + 1 & otherwise \end{cases} \tag{30}$$

where $Y = (y_1, y_2, ..., y_n)$ is the groud truth of reponses. $Object(\cdot)$ is function to extract the object of triplet. The loss of three terms is calculated as:

$$\mathcal{L}_g = -\sum_{m=1}^{b+T} \hat{g}_m \log g_m + (1 - \hat{g}_m) \log(1 - g_m) \tag{31}$$

$$\mathcal{L}_l = -\sum_{t=1}^{n} -\hat{l}_t \log P_t^{kb} \tag{32}$$

$$\mathcal{L}_v = -\sum_{t=1}^{n} -\hat{y}_t \log P_t^{vocab} \tag{33}$$

Then we sum the three terms up to get $\mathcal{L}_{basic}$. You can find more details about the global-to-local pointer mechanism in Wu et al. (2019).

A.2 Dataset

We follow the partition as Madotto et al. (2018); Wu et al. (2019) on In-Car Assistant and Qin et al. (2020) on Multi-WOZ 2.1. The details are about two dataset as tabel 5

Dataset	**In-Car Assistant**		
Vocab size	1651		
Avg. dialog turns	2.6		
Avg. length of sent.	8.1		
Domain	**Navigate**	**Weather**	**Schedule**
Dialogs	1000	996	1035
Partition	**Train**	**Dev**	**Test**
	2425	302	304
Dataset	**Multi-Woz2.1**		
Vocab size	3725		
Avg. dialog turns	4.6		
Avg. length of sent.	14.4		
Domain	**Restaurant**	**Attraction**	**Hotel**
Dialogs	1309	150	635
Partition	**Train**	**Dev**	**Test**
	1839	117	141

Table 5: Details of two multi-domain dataset.

A.3 Hyper-parameters

We set the encoder-decoder hidden size from {100,200} and signature block from {25, 50, 100}. We use the *glove.6b* word vector to initialize the embedding matrix of the encoder and decoder. Then we random initialize the embedding matrix of the memory network. For tokens with '_', we first split them into a token list and use the BOW of word vectors. We adopt an exponential schedule sampling for decoding, and the schedule is calculated as He et al. (2020a):

$$tfr = \frac{\alpha}{\alpha + e^{\frac{epoch}{\alpha}} - 1} \tag{34}$$

where tfr is sampling probability from ground truth. We set α from $\{10, 15, 20\}$. For adjacent n-gram patterns, we set n from $\{2, 3, 4\}$.

A.4 Experiment details

For the **main experiment** results, we adopt the reported results of baselines except *DA-HIMN*.

For the **ablation study**, we train our model on the hyper-parameter set to get the best result of different signatures.

For the **domain adaption experiment** results, we rerun the code of *DF-net* and *GLMP*. To avoid the influence of model dimensions and pre-trained word vectors on the experimental results, we adopt the same model dimensions (200d) and GloVe vectors (*glove.6B.200d*) for our model and the baseline model. Word vectors are used in the same way as in A.3

For **model scale**, we add the Multi-WOZ2.1 dataset based on In-Car Assistant to initialize the lang. We calculate the model size in the same model dimen-

sion setting and compute the size growth on average of the three domains of Multi-WOZ2.1.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 6

☑ A2. Did you discuss any potential risks of your work?
section 7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

section 3.2

☑ B1. Did you cite the creators of artifacts you used?
section 3.2

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
section 7

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We use pre-trained word vector in our work and our use is consistent with their intended use.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. We didn't use any data at risk in our work

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In the footnote

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix

C ☑ Did you run computational experiments?

section 3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
section 3.5.3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
section 3.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
section 2.1

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.