Neurocomputing 440 (2021) 279–286

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Self-supervised attention flow for dialogue state tracking

Boyuan Pan [a,*,1], Yazheng Yang [b,1], Bo Li [c], Deng Cai [a,d]

[a] State Key Lab of CAD&CG, Zhejiang University, China
[b] Computer Science and Technology, Zhejiang University, China
[c] University of Illinois Urbana-Champaign, United States
[d] Alibaba-Zhejiang University Joint Institute of Frontier Technologies, China

ARTICLE INFO

Article history:
Received 13 August 2020
Revised 15 January 2021
Accepted 30 January 2021
Available online 18 February 2021
Communicated by Zidong Wang

Keywords:
Self-supervised learning
Dialogue state tracking
Attention mechanism

ABSTRACT

The performance of existing approaches for dialogue state tracking (DST) is often limited by the deficiency of labeled datasets, and inefficient utilization of data is also a practical yet tough problem of the DST task. In this paper, we aim to tackle these challenges in a self-supervised manner by introducing an auxiliary pre-training task that learns to pick up the correct dialogue response from a group of candidates. Moreover, we propose an attention flow mechanism that is augmented with a soft-threshold function in a dynamic way to better understand the user intent and filter out the redundant information. Extensive experiments on the multi-domain dialogue state tracking dataset MultiWOZ 2.1 demonstrate the effectiveness of our proposed method, and we also show that it is able to adapt to zero/few-shot cases under the proposed self-supervised framework.

1. Introduction

Dialogue State Tracking (DST) is a core component of task-oriented dialogue systems, such as ticket booking or restaurant reservation. The goal of DST is to identify user's intention expressed during a conversation in the form of dialogue states, which are a set of slot and value pairs. For example, as shown in Table 1, the dialogue states extracted from the conversation are *(restaurant-food, traditional)*, *(restaurant-pricerange, moderate)*, *etc.* The performance of DST is crucial to measuring the dialogue management, where user intention determines the next system action and/or the content to query from the databases [1].

Traditional DST models are mostly based on pre-defined ontology lists that specify all possible slot values in advance [2–4]. It helps to simplify DST into a classification problem which generates a score for each candidate of *(slot, value)*. However, in real-world scenarios, it is not feasible to gain access to enumerate all the possible values from a large dynamically changing knowledge base [5]. Recently, many DST works focus on generative models that generate values from open-vocabulary or copy entities from the dialogue history and gain great popularity [1,6,7].

Although these trackers obtain significant success, they are still limited by the insufficient amount of annotated data which is lack of diversity. Moreover, the data-hungry nature of the neural networks makes the existing DST models difficult to generalize well to the scenarios with sparse data [8], where only a few slots in all candidates are targeted in a single dialogue turn.

In this paper, we present a novel framework named *Self-Supervised Attention Flow* (SAF) network for dialogue state tracking. We introduce a self-supervised learning task, Dialogue Response Selection (DRS), to help guide meaningful and attentional dialogue learning. The task is defined as, given the dialogue history and all possible slots, the model is required to predict the next system response from a set of candidates. To select the correct response, especially for task-oriented dialogue settings, the DRS model needs to understand the intent of the user and integrate all the information of the slots and response candidates, which is an intuitive transferring knowledge source for dialogue state tracking. Without the need of any annotated data, the DRS model can learn to understand the sentence-level relationships in the same knowledge domain, thus greatly improves the utilization efficiency of the given dataset. Moreover, we employ a *soft-threshold* function to dynamically update the encoding representation to form an attention flow mechanism in order to filter out the redundant input information. Our contributions can be summarized as follows:

* Corresponding author.
E-mail addresses: panby@zju.edu.cn (B. Pan), yazheng_yang@zju.edu.cn (Y. Yang), lbo@illinois.edu (B. Li), dengcai@cad.zju.edu.cn (D. Cai).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.neucom.2021.01.118
0925-2312/

Table 1
An example of multi-domain dataset, MultiWOZ 2.1. At each turn, the DST needs to track the slot values *(domain-slot, value)* mentioned by the user for all the possible slots. The dialogue state is accumulated as the dialogue proceeds.

User: I am going to Cambridge and interested in trying some restaurants. Can you recommend one that serves traditional food? **Dialogue State**: *(restaurant-food, traditional)*
System: Please provide more information to help us serve you better. **User**: Just something in the moderate price range is all I care about really. **Dialogue State**: *(restaurant-food, traditional), (restaurant-pricerange, moderate)*
System: My apologies. There is nothing in the moderate price range that is traditional. Would you like to try a different type of restaurant? **User**: How about modern European food? **Dialogue State**: *(restaurant-pricerange, moderate), (restaurant-food, modern european)*

- We introduce the task of dialogue response selection (DRS), which is under a Self-Supervised Attention Flow (SAF) framework to learn the potential information and dependencies in the dialogue and domain/slot entities.
- We incorporate a novel attention flow mechanism that leverages a simple yet effective soft-threshold function to capture key information from the sparse data.
- Our SAF framework achieves the state-of-the-art result on the recently released multi-domain task-oriented dialogue dataset MultiWOZ 2.1 [9,10]. We also show that our model enables zero/few-shot learning where no (or only a few) annotated examples in the training time are available.

2. Methodology

Our Self-Supervised Attention Flow (SAF) model is composed of two parts: dialogue response selection (DRS) and dialogue state tracking (DST). We first pre-train the DRS model (shown in Fig. 1) and then transfer the prior knowledge learned by the DRS model to the DST model.

2.1. Dialogue response selection

Dialogue response selection requires to pick up the most appropriate response from several candidates given the dialogue history and all the possible slots. To collect the candidates, we simply mix the correct response sentence and four random system responses (as negative samples) from other dialogues, which requires no extra human annotations and can also make full use of the given dataset.

Formally, the inputs of DRS includes the dialogue history $X = (x_1, x_2, \dots x_N)$, where N is the length of the text; the sequence of all applicable *(domain, slot)* pairs $Z = ((d_1, s_1), \dots, (d_G, s_H))$, where G and H are the total numbers of domains and slots; and the concatenation of response candidates $X^r = (x_1^r, x_2^r, \dots, x_M^r)$, where M is the length of the text, and the candidates are separated by a signal <SEP>. The goal is to select the correct response from the five candidates.

Encoding Layer. For the dialogue history X and response candidates X^r, we use a trainable token-level embedding layer and layer normalization [11] to encode the text. We also follow [12] to apply a positional encoding layer with sine and cosine functions to the embeddings. The dimension of both embeddings are the same, so that the two can be element-wise summed. The final embeddings of X and X^r are:

$$\mathbf{U} = \mathbf{U}_{emb} + PE(X) \in \mathbb{R}^{N\times d} \mathbf{U}^r = \mathbf{U}^r_{emb} + PE(X^r) \in \mathbb{R}^{M\times d} \tag{1}$$

For the *(domain, slot)* pairs Z, we follow [7] to use two separate embeddings to encode the domains and slots, and then combine them by element-wise summation. Formally:

$$\mathbf{u}^z_{d_g,s_h} = \mathbf{u}^z_{d_g} + \mathbf{u}^z_{s_h} \in \mathbb{R}^d \mathbf{U}^z = \mathbf{u}^z_{d_1,s_1} \oplus \cdots \oplus \mathbf{u}^z_{d_G,s_H} \in \mathbb{R}^{K\times d} \tag{2}$$

where $\oplus$ denotes concatenation, K is the total number of *(domain, slot)* pairs.

Attention Flow Layer. Attention mechanism is responsible for linking and fusing information from the given knowledge, and the recent developments of attention mechanism have shown its impressive performance in many NLP tasks [12–14]. However, oftentimes the dialogue history is informative and complicated, and one single layer of attention may be insufficient to comprehend the subtle relationship among the context and the knowledge of given domains and slots. Moreover, for each single turn, the useful *(domain, slot)* pairs in Z is usually very sparse, so it is important to filter out the redundant information. Therefore, we propose the attention flow layers linked up with a *soft-threshold* function, which has often been used as a key step in many signal denoising methods [15,16], to iteratively update the encoding representation. Given the encoded dialogue history $\mathbf{U}$, response candidates $\mathbf{U}^r$, and *(domain, slot)* pairs $\mathbf{U}^z$, we adopt the multi-head attention [12] to project the representations into multiple sub-spaces. The attention is computed by scaled dot-product operations between query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$:

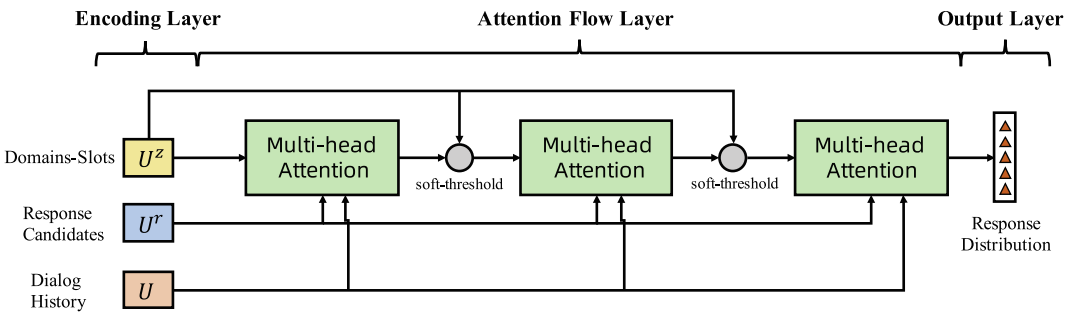


Fig. 1. Overview of our dialogue response selection (DRS) model, comprising an encoding layer, a multi-hop attention flow layer (three hops in the figure) and an output layer.

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{3}$$

where d_k is the dimension of keys. We use three attention layers to learn the potential dependencies between $\mathbf{U}, \mathbf{U}^r$, and $\mathbf{U}^z$:

$$\begin{aligned}
\mathbf{H}_{a1} &= \text{Attention}(\mathbf{U}^z,\mathbf{U}^z,\mathbf{U}^z) \in \mathbb{R}^{K\times d}\\
\mathbf{H}_{a2} &= \text{Attention}(\mathbf{H}_{a1},\mathbf{U}^r,\mathbf{U}^r) \in \mathbb{R}^{K\times d}\\
\mathbf{H}_{a3} &= \text{Attention}(\mathbf{H}_{a2},\mathbf{U},\mathbf{U}) \in \mathbb{R}^{K\times d}
\end{aligned} \tag{4}$$

For simplicity, we denote the attention Eqs. (4) as $\mathbf{H}_{a3} = F(\mathbf{U}^z,\mathbf{U}^r,\mathbf{U})$ and do not express the multi-head attention, and we refer the readers to [12] for more details. To eliminate noise-related information and construct highly discriminative features, we send the result $\mathbf{H}_{a3}$ to a soft-threshold function, which is typically defined as:

$$g(x) = \begin{cases} x-\tau, & x > \tau \\ 0, & -\tau \leqslant x \leqslant \tau \\ x+\tau, & x < -\tau \end{cases} \tag{5}$$

where x is the input feature, τ is the threshold, *i.e.*, a positive parameter. Instead of setting the negative features to zero in the ReLU activation function, soft-thresholding sets the near-zero features to zeros, so that useful negative features can be preserved. We set τ as a vector, where each dimension handles the features of the corresponding *(domain, slot)* pair:

$$\begin{aligned}
&\tau = \alpha \odot \text{mean}(|\mathbf{H}^{res}|) \in \mathbb{R}^K \quad \mathbf{H}^{res} = \mathbf{H}_{a3} + \mathbf{U}^z \in \mathbb{R}^{K\times d}\\
&\alpha = \text{sigmoid}(\mathbf{H}^{res}\cdot\mathbf{w}_1) \in \mathbb{R}^K
\end{aligned} \tag{6}$$

where $\odot$ is element-wise product, mean denotes row-wise mean pooling, $\mathbf{w}_1$ is a trainable parameter, $\mathbf{H}^{res}$ is a residual connection around $\mathbf{U}^z$ and output of the F. Note that the subtraction in Eq. (5) is operated by expanding τ for d times as a matrix.

We combine $\mathbf{U}^z$ and the output of the soft-threshold $g(\mathbf{H}_{a3})$ by $\mathbf{H}_z^{(2)} = \mathbf{W}_c[\mathbf{U}^z; g(\mathbf{H}_{a3})] \in \mathbb{R}^{K\times d}$ as the representation of Z to be used in the next flow of attention F:

$$\begin{aligned}
\mathbf{H}_{a1}^{(2)} &= \text{Attention}(\mathbf{H}_z^{(2)},\mathbf{U}^z,\mathbf{U}^z) \in \mathbb{R}^{K\times d}\\
\mathbf{H}_{a2}^{(2)} &= \text{Attention}(\mathbf{H}_{a1}^{(2)},\mathbf{U}^r,\mathbf{U}^r) \in \mathbb{R}^{K\times d}\\
\mathbf{H}_{a3}^{(2)} &= \text{Attention}(\mathbf{H}_{a2}^{(2)},\mathbf{U},\mathbf{U}) \in \mathbb{R}^{K\times d}
\end{aligned} \tag{7}$$

This iterative procedure halts when a maximum number of reasoning layers T is reached ($T \geqslant 1$). The final attention representation $\mathbf{H}_{a3}^{(T)}$ is fed into the output layer.

Output Layer. The DRS task requires the model to select the correct response from the given candidates. We obtain the probability distribution by a linear function with a softmax function:

$$\mathbf{p}_{drs} = \text{softmax}(\mathbf{w}_p^\top \mathbf{H}_{a3}^{(T)} \mathbf{w}_q) \tag{8}$$

where $\mathbf{w}_p, \mathbf{w}_q$ are trainable parameters, $\mathbf{p}_{drs}$ is a 5-dimensional probability distribution.

2.2. Dialogue state tracking

As presented in Fig. 2, we show how our Self-Supervised Attention Flow can be applied to the dialogue state tracking scenario.

We adopt the architecture of [7], a non-autoregressive DST model that enables decoding dialogue states in parallel rather than the traditional sequence-to-sequence schema. Given the dialogue history X and domain-slots background Z, we train two models in parallel: (1) fertility generation model and (2) state generation model. The fertility generation model first decides the length of the final states by selecting the domain-slot pairs from Z, and then the state generation model can simultaneously predict the value of each domain-slot pair.

Fertility Generation Model. The "fertility" of a *(domain, slot)* pair is the length of its value tokens [17]. For example, if the value of the pair *(taxi, destination)* is "*De Luca Cucina and Bar*", then its fertility is 5. Note that the predicted fertility vector is supposed to be sparse since most of the *(domain, slot)* pairs are irrelevant to the dialogue history and their fertilities should be 0.

Following [18,7], we also add a partially delexicalized dialogue history X_d to the input of the model. The dialogue history is delexicalized till the last system utterance by removing real-value tokens that match the previously decoded slot values to tokens expressed as *domain_slot*. For example, the user utterance "I would like to leave by 13:00" is delexicalized to "I would like to leave by *taxi_leaveat*". This approach makes use of the previous predicted states from the DST model and doesn't rely on extra knowledge. Formally, we have

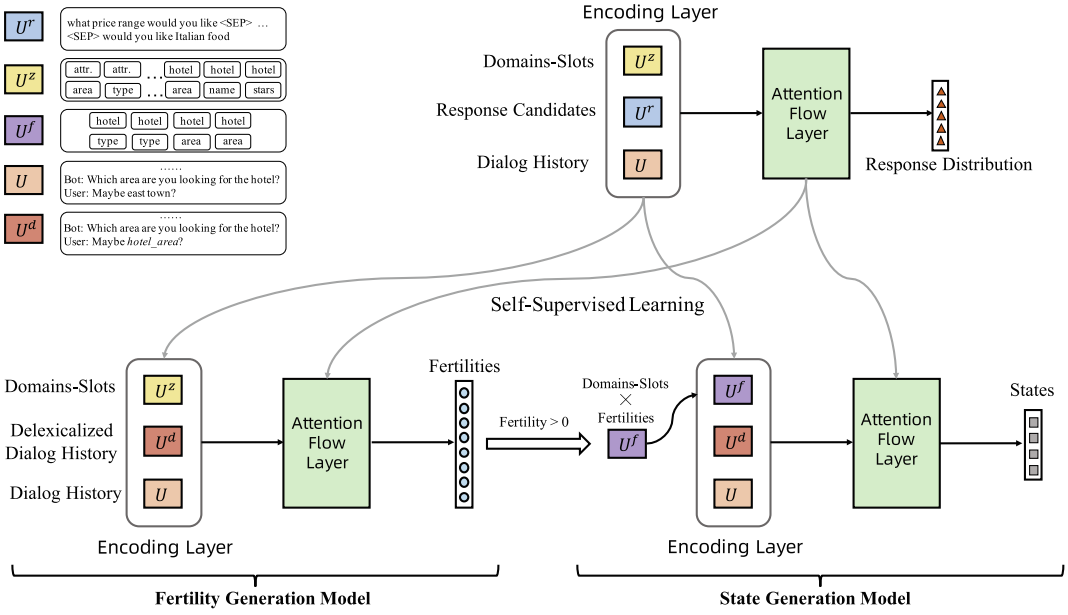


Fig. 2. Overview of our Self-Supervised Attention Flow (SAF) framework. The upper part is the dialogue response selection model, the bottom part is the dialogue state tracking model (consists of a fertility generation model and a state generation model) to which the learned knowledge will be transferred.

$$\mathbf{H}_f^{(T)} = G_f(\mathbf{U}^z, \mathbf{U}^d, \mathbf{U}) \in \mathbb{R}^{K\times d} \tag{9}$$

where G_f is the fertility generation model, $\mathbf{U}^d$ is the encoding representation of X_d, $\mathbf{H}_f^{(T)}$ is the final attention representation as $\mathbf{H}_{a3}^{(T)}$ in the DRS task. G_f shares the same framework with the encoding layer and attention flow layer of DRS model except replacing $\mathbf{U}^r$ with $\mathbf{U}^d$. For the output distribution, we have

$$\mathbf{P}_f = \text{softmax}(\mathbf{H}_f^{(T)}\mathbf{W}_f) \in \mathbb{R}^{K\times n} \tag{10}$$

where $\mathbf{W}_f$ is a trainable parameter, n is the maximum fertility in the dataset. We use the cross-entropy objective to train the fertility generation model.

Therefore, we can obtain a K-dimension vector that each dimension indicates the fertility Y_{d_g,s_h} of a domain-slot pair (d_g, s_h). To form the input of the state generation model for non-autoregressive decoding, we repeat each (d_g, s_h) pair for Y_{d_g,s_h} times and concatenate them sequentially: $X_f = [(d_1,s_1)^{Y_{d_1,s_1}}; \ldots; (d_G,s_H)^{Y_{d_G,s_H}}]$, where the length of X_f is $N_Y = \sum_{g=1,h=1}^{K} Y_{d_g,s_h}$.

State Generation Model. Given the dialogue history X, delexicalized dialogue history X_d, and the input domain-slot sequence X_f, we apply the same model structure as used in the fertility generation model:

$$\mathbf{H}_s^{(T)} = G_s(\mathbf{U}^f, \mathbf{U}^d, \mathbf{U}) \in \mathbb{R}^{N_Y\times d} \tag{11}$$

where G_s is the state generation model, $\mathbf{U}^f$ is the encoding representation of X_f, $\mathbf{H}_s^{(T)}$ is the final attention representation. The output distribution can be decoded by:

$$\mathbf{P}_s^{vocab} = \text{softmax}(\mathbf{H}_s^{(T)}\mathbf{W}_{vocab}) \in \mathbb{R}^{N_Y\times V} \tag{12}$$

where $\mathbf{W}_{vocab}$ is the trainable parameter, V is the size of the vocabulary. As open-vocabulary DST models do not require a given slot ontology, We also incorporate a pointer network [19,20] to copy words from the dialogue history. Now the output distribution becomes:

$$\mathbf{P}_s = \lambda \times \mathbf{P}_s^{vocab} + (1-\lambda) \times \mathbf{P}_s^{pt} \in \mathbb{R}^{N_Y\times V} \tag{13}$$

where $\mathbf{P}_s^{pt} = \text{softmax}(\mathbf{H}_s^{(T)}, \mathbf{U})$ is the probability of copying words from X. λ is the weight to balance the two:

$$\lambda = \text{sigmoid}(\mathbf{w}_s[\mathbf{H}_s^{(T)}; \mathbf{U}^f; \tilde{\mathbf{U}}]) \in \mathbb{R}^{N_Y} \tag{14}$$

where $\tilde{\mathbf{U}}$ is transformed representation of $\mathbf{U}$ to match the dimensions of $\mathbf{H}_s^{(T)}$. We use the cross-entropy objective to train the state generation model.

Self-Supervised Learning. As mentioned at the beginning of Section 2, before training the DST model, we first pre-train a DRS model. Since the DRS model has the ability to predict the correct next response, it has potential in semantic understanding and can capture the deep dependencies between the dialogue history and the knowledge of the slots.

We initialize both G_f of the fertility generation model (Eq. (9)) and G_s of the state generation model (Eq. (11)) by the corresponding modules of the pre-trained DRS model, which are the combination of the encoding layer and attention flow layer. For the task of DRS, because one correct response can be mixed with several negative sample groups, the size of training data can be much larger than it is in the DST task, thus improves the robustness of the DRS model. In this way, given the prior knowledge of the DRS model, our DST model has more advantages in capturing the sentence-level information and understanding the intention of the user.

3. Experiments

3.1. Datasets

MultiWOZ 2.1 [10] is a recently released large multi-domain dialogue dataset spanning seven distinct domains and containing over 10,000 dialogues. We identified 5 domains, 30 domain-slot pairs and over 4,500 possible values, which is much larger than existing standard datasets like WOZ [21] and DSTC2 [22]. We pre-processed the dialogues by tokenizing, lower-casing, and delexicalizing all system responses following the pre-processing scripts from [1].

3.2. Implementation details

We set the embedding dimension as 256 in all experiments and fix the number of attention heads to 16 in all attention layers. We shared the embedding weights to embed domain and slot tokens as input to the fertility generation model and state generation model. We also shared the embedding weights between dialogue history encoder and state generation model. We set the dropout [23] ratio as 0.2. We adopt the Adam optimizer [24] and the learning rate strategy similarly as [12]. We set the hops of attention flow as 3. During DST training, we adopt the teacher-forcing learning strategy as in [7] to use the ground-truth fertility as input to the state generation model and the ground-truth delexicalized dialogue history as the input to both models of DST. During the labels collection of DRS, for each gold response, we randomly sample 4 negative responses for 5 times, and mix the gold response with these 5 groups respectively.

3.3. Evaluation metrics

We evaluate model performance by the joint goal accuracy as commonly used in DST [22,7]. The joint goal accuracy compares the predicted belief states to the ground truth at each turn. The joint accuracy is 1.0 if and only if all *(domain, slot, value)* triplets are predicted correctly at each turn, otherwise 0. In ablation studies, we also use the slot accuracy, which individually compares each *(domain, slot, value)* to its ground truth label.

3.4. Baselines

HJST & *FJST.* [10]. FJST refers to Flat Joint State Tracker, which consists of a dialogue history encoder as a bidirectional LSTM network. HJST follows a similar architecture but uses a hierarchical LSTM network [25] to encode the dialogue history.

DST Reader. [26]. It treats dialogue state tracking as a reading comprehension problem. Given the dialogue history, it learns to extract slot values as spans.

HyST. [27]. It combines a hierarchical encoder in a fixed vocabulary system with an open vocabulary n-gram copy-based system.

TRADE. [1]. This model makes use of a pointer network with index-based copying instead of a token-based copying mechanism.

NADST. [7]. This is the current state-of-the-art model on the Multi-WOZ 2.1 dataset. It uses a Transformer-based non-autoregressive decoder to generate the current turn dialogue state.

3.5. Results and analysis

In Table 2, we compare our model with other competitive published models on MultiWOZ 2.1. As we can see, our method Self-supervised Attention Flow (SAF) network achieves 51.6% of joint goal accuracy, which clearly outperforms all the baselines and achieves the state-of-the-art result.

Table 2
DST joint accuracy on MultiWOZ 2.1 dataset. †: results reported by [10].

Model	Joint Acc
HJST [10]†	0.356
DST Reader [26]†	0.364
FJST [10]†	0.378
HyST [27]†	0.381
TRADE [1]†	0.453
NADST [7]	0.490
SAF (Ours)	**0.516**

Table 3
Ablation analysis on MultiWOZ 2.1 dataset. "Joint" denotes joint goal accuracy, "Slot" denotes slot accuracy."T" is the hops of attention flow.

Model	Joint	Slot
No Fertility Model Transfer	0.498	0.971
No State Model Transfer	0.504	0.972
No Soft-threshold	0.507	0.973
No Soft-threshold (ReLU)	0.506	0.973
SAF ($T = 1$)	0.446	0.968
SAF ($T = 2$)	0.474	0.970
SAF ($T = 3$)	0.516	0.975

We also conduct an ablation experiment to evaluate the individual contribution of each component of our model, as shown in Table 3. Firstly we randomly initialize G_f the fertility generation model, which means we don't transfer the parameters of it from the pre-trained DRS model. We observe that the performance drops significantly to 0.498 on joint accuracy and 0.971 on slot accuracy, which indicates that the dialogue response selection has deep connections with dialogue state tracking in high-level semantic space. Similarly, we then ablate the transfer of G_s in the state generation model, the result also demonstrates the effectiveness of our self-supervised learning scheme by this auxiliary task. We also ablate the soft-threshold function and try: (1) directly using the input of the soft-threshold function as the representation of Z in the next flow of attention and (2) using ReLU activation function to replace it. Both results on joint accuracy drop about 0.10, and we can conclude that the soft-threshold function has its own advantage in denoising and filtering out the redundant information. Finally, when we vary our models for different hops, we found that our SAF obtains the best performance when $T = 3$, which indicates that the attention flow structure can improve the deep reasoning for the dialogue history and the slot values.

3.6. Zero/few-shot learning

In Table 4, we show the ability of our SAF model to generalize to unseen domains by running zero-shot and few-shot learning experiments. In the zero-shot setting, we exclude a specific domain in the training data, while in the few-shot case, we keep only 1% of the original training data in that specific domain. As we can see, both the baseline model and our SAF model drop a lot on each domain without sufficient training data. However, compared to the baseline model, our SAF model significantly improves the performance on all of the domains, especially "attraction", "hotel" and "restaurant". For the reason of the insensitivity of "taxi" and "train", we conjecture this is because these two domains have many similar slots and values (e.g., *destination, arriveby*) so that they can learn from each other when the training data of one is missing. These results demonstrate that our self-supervised scheme on the dialogue response selection task learns substantial prior knowledge that has deep relations with DST, which will be more effective when larger amounts of unlabeled dialogue context data are available.

3.7. Visualization

In Fig. 3, we also present a visualized analysis of self-attention scores of four heads in the state generation model. We can see that the values are highly correlated between the different domains which have the similar slots such as *hotel-bookday* with *train-day*, *hotel-bookpeople* with *train-bookpeople*. We can also observe that the tokens within the same *domain-slot* have strong connections with each other, which may improve the coherency and integrity of the entities.

3.8. Error analysis

In Table 5, we show two examples of typical errors. In the first case, the ground truth value of *train-destination* is *London*, but the model's prediction is *London Kings Cross*. We observe that "London Kings Cross" is a phrase that appears very frequently in the corpus, which indicates that if we expand the dataset too many times in the pre-training procedure, the model may suffer from overfitting. The second example shows a problem of knowledge and language understanding. Although successfully predicting the departure address, the model still recognizes a part of the address as the departure time in both turns. We conjecture that there might be a saddle point between the hops of the attention flow and the text understanding depth of the model, and we believe commonsense knowledge is necessary for further improvement of machine comprehension.

4. Related works

4.1. Dialogue state tracking

At early stage, dialogue state trackers combine semantic information extracted by Natural Language Understanding (NLU) modules to update the current dialogue states turn by turn [28–31]. Subsequent approaches combine NLU and DST to remove the need of NLU and thus reduce the accumulated errors and the credit assignment problems [32,33]. Within the body of this research, a great promotion had been made by deep learning models [21,18,3,4], which represent the dialogue state as a distribution over all candidate slot values that are defined in the ontology. However, these methods rely heavily on a pre-defined and comprehensive ontology, which is usually impractical in the real world application.

Recently, open-vocabulary approaches based on generative models have been increasingly studied. [18] propose a dialogue model consisting of an RNN encoder and two RNN decoder with a pointer network. [27] combine both fixed-vocabulary and open-vocabulary approaches by separately choosing which approach is more suitable for each slot. [1] integrate a slot gating module and copy mechanism to generate slot values in states. [7] use a Transformer-based non-autoregressive decoder to generate the current turn dialogue state.

Nevertheless, very few works focus on alleviating the problems of data deficiency and sparsity. [8] proposed an RL-based data augmentation method to generate training data for DST. In this paper, we introduce an auxiliary task to improve the utilization efficiency of the dataset and leverage a self-supervised learning scheme to transfer the knowledge.

4.2. Self-supervised learning

Self-supervised learning, which automatically constructs some supervisory signals directly computed from the unlabeled data,

Table 4
Joint goal accuracy and slot accuracy on zero-shot and few-shot MultiWOZ 2.1 experiments. "Baseline" has the same model structure with our SAF but without the self-supervised learning procedure. "0%" denotes zero-shot learning, "1%" denotes few-shot learning, and "100%" denotes training with the full dataset.

Models	Attraction		Hotel		Restaurant		Taxi		Train	
	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot
Baseline (0%)	0.245	0.934	0.144	0.889	0.128	0.878	0.542	0.967	0.273	0.938
SAF (0%)	**0.393**	**0.958**	**0.315**	**0.950**	**0.303**	**0.948**	**0.614**	**0.972**	**0.341**	**0.952**
Baseline (1%)	0.284	0.941	0.179	0.912	0.169	0.895	0.572	0.969	0.301	0.943
SAF (1%)	**0.433**	**0.960**	**0.345**	**0.955**	**0.341**	**0.954**	**0.635**	**0.974**	**0.387**	**0.955**
SAF (100%)	0.640	0.991	0.384	0.967	0.458	0.976	0.707	0.987	0.545	0.975

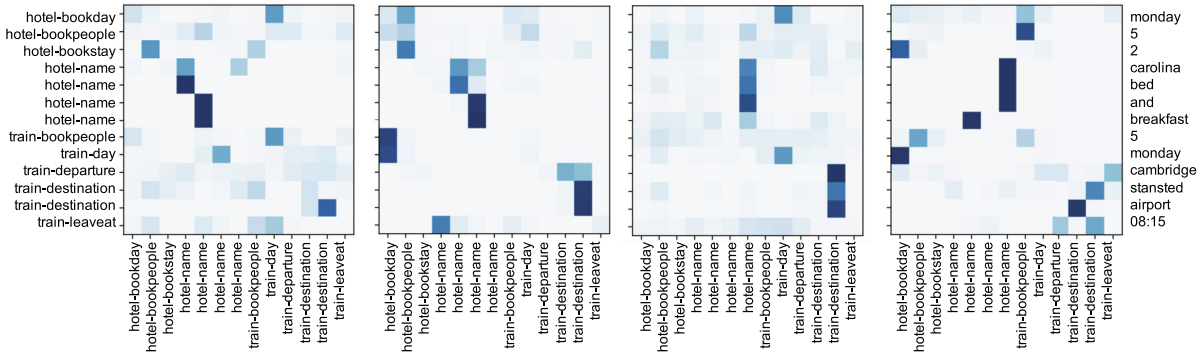


Fig. 3. Visualization of self-attention scores of four heads in the state generation model. The corresponding prediction output for each representation is presented on the right side.

Table 5
Error Analysis. Two examples (seperated by double lines) generated by our SAF model on the MultiWOZ 2.1 dataset.

User: I need a train to London please.
Gold Dialogue State: *(train-destination, London)*
Predicted Dialogue State: *(train-destination, London Kings Cross)*
User: I am looking to book a taxi that will leave from the missing sock.
Gold Dialogue State: *(taxi-departure, the missing sock)*
Predicted Dialogue State: *(taxi-departure, the missing sock), (taxi-leaveat, missing)*

System: What is your destination?
User: I need to get to the lucky star please.
Gold Dialogue State: *(taxi-departure, the missing sock), (taxi-destination, the lucky star)*
Predicted Dialogue State: *(taxi-departure, the missing sock), (taxi-destination, the lucky star), (taxi-leaveat, lucky)*

has been successfully applied in computer vision such as image completion [34], image colorization [35–37] and channel prediction [38]. As for natural language processing, recent developments of large-scale pre-trained language models have demonstrated their dominant performances in many down-stream NLP tasks [39–41]. [42] propose the sentence-level modeling for extractive summarization. [43] consider sequential order as the self-supervised signal in dialogue generation. [44] mine supervision information in the attention mechanism for sentiment analysis. Our work considers the dialogue response selection as the self-supervised signal in DST and boosts the ability of sentence-level information capturing and user intention understanding.

5. Conclusion

In this paper, we propose *Self-Supervised Attention Flow* network for the task of dialogue state tracking. We transfer the knowledge learned from a simple auxiliary task called dialogue response selection to alleviate the problems of data deficiency and sparsity. Moreover, we propose an attention flow mechanism augmented by a soft-threshold function to filter out the redundant information and dynamically update the embedding representations. The experiments show that our method achieves the state-of-the-art result on the MultiWOZ dataset and its potential zero/few-shot ability. Future works involve methods of more efficient utilization of data and exploration in the detailed dependencies among user intention, system actions, and the slot-value ontology.

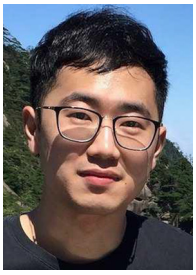
Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C.-S. Wu, A. Madotto, E. Hosseini-Asl, C. Xiong, R. Socher, P. Fung, Transferable multi-domain state generator for task-oriented dialogue systems, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 808–819.
[2] V. Zhong, C. Xiong, R. Socher, Global-locally self-attentive dialogue state tracker, arXiv preprint arXiv:1805.09655..
[3] O. Ramadan, P. Budzianowski, M. Gasic, Large-scale multi-domain belief tracking with knowledge sharing, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2018, pp. 432–437..

[4] H. Lee, J. Lee, T.-Y. Kim, Sumbt, Slot-utterance matching for universal and scalable belief tracking, in, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5478–5483.
[5] P. Xu, Q. Hu, An end-to-end approach for handling unknown slot values in dialogue state tracking, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 1448–1457..
[6] L. Ren, J. Ni, J. McAuley, Scalable and accurate dialogue state tracking via hierarchical sequence generation, 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), 2019, pp. 1876–1885.
[7] H. Le, R. Socher, S.C. Hoi, Non-autoregressive dialog state tracking, in: International Conference on Learning Representations, 2020..
[8] Y. Yin, L. Shang, X. Jiang, X. Chen, Q. Liu, Dialog state tracking with reinforced data augmentation, AAAI..
[9] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, M. Gasic, 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, 2018, pp. 5016–5026.
[10] M. Eric, R. Goel, S. Paul, A. Sethi, S. Agarwal, S. Gao, D. Hakkani-Tur, Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines, arXiv preprint arXiv:1907.01669..
[11] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450..
[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008..
[13] M. Seo, A. Kembhavi, A. Farhadi, H. Hajishirzi, Bidirectional attention flow for machine comprehension, ICLR..
[14] H.-Y. Huang, E. Choi, W. tau Yih, FlowQA: Grasping flow in history for conversational machine comprehension, in: International Conference on Learning Representations, 2019..
[15] D.L. Donoho, De-noising by soft-thresholding, IEEE Transactions on Information Theory 41 (3) (1995) 613–627.
[16] K. Isogawa, T. Ida, T. Shiodera, T. Takeguchi, Deep shrinkage convolutional neural network for adaptive noise reduction, IEEE Signal Processing Letters 25 (2) (2017) 224–228.
[17] J. Gu, J. Bradbury, C. Xiong, V.O. Li, R. Socher, Non-autoregressive neural machine translation, in: International Conference on Learning Representations, 2018..
[18] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, D. Yin, 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), 2018, pp. 1437–1447..
[19] O. Vinyals, M. Fortunato, N. Jaitly, Pointer networks, in: Advances in Neural Information Processing Systems, 2015, pp. 2692–2700..
[20] A. See, P.J. Liu, C.D. Manning, Get to the point: Summarization with pointer-generator networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Long Papers, 1, 2017, pp. 1073–1083.
[21] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gasic, L.M.R. Barahona, P.-H. Su, S. Ultes, S. Young, A network-based end-to-end trainable task-oriented dialogue system, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, 2017, pp. 438–449..
[22] M. Henderson, B. Thomson, J.D. Williams, The second dialog state tracking challenge, in: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), 2014, pp. 263–272.
[23] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (1) (2014) 1929–1958.
[24] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980..
[25] I.V. Serban, A. Sordoni, Y. Bengio, A. Courville, J. Pineau, Building end-to-end dialogue systems using generative hierarchical neural network models, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016..
[26] S. Gao, A. Sethi, S. Agarwal, T. Chung, D. Hakkani-Tur, A.A. AI, Dialog state tracking: A neural reading comprehension approach, in: 20th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2019, p. 264.
[27] R. Goel, S. Paul, D. Hakkani-Tür, Hyst: A hybrid approach for flexible and accurate dialogue state tracking, arXiv preprint arXiv:1907.00883..
[28] J.D. Williams, S. Young, Partially observable markov decision processes for spoken dialog systems, Computer Speech & Language 21 (2) (2007) 393–422.
[29] J.D. Williams, Web-style ranking and slu combination for dialog state tracking, in, in: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), 2014, pp. 282–291.
[30] G. Kurata, B. Xiang, B. Zhou, M. Yu, Leveraging sentence-level information with encoder LSTM for semantic slot filling, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016.
[31] Y. Shi, K. Yao, H. Chen, D. Yu, Y.-C. Pan, M.-Y. Hwang, Recurrent support vector machines for slot tagging in spoken language understanding, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 393–399..
[32] M. Henderson, B. Thomson, S. Young, Word-based dialog state tracking with recurrent neural networks, in: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), 2014, pp. 292–299.
[33] N. Mrkšić, D.Ó. Séaghdha, T.-H. Wen, B. Thomson, S. Young, 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), 2017, pp. 1777–1788.
[34] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: Feature learning by inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.
[35] R. Zhang, P. Isola, A.A. Efros, Colorful image colorization, in: European Conference on Computer Vision, Springer, 2016, pp. 649–666.
[36] G. Larsson, M. Maire, G. Shakhnarovich, Learning representations for automatic colorization, in: European Conference on Computer Vision, Springer, 2016, pp. 577–593.
[37] G. Larsson, M. Maire, G. Shakhnarovich, Colorization as a proxy task for visual understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6874–6883.
[38] R. Zhang, P. Isola, A.A. Efros, Split-brain autoencoders: Unsupervised learning by cross-channel prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1058–1067.
[39] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805..
[40] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners..
[41] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: Advances in Neural Information Processing Systems, 2019, pp. 5754–5764..
[42] H. Wang, X. Wang, W. Xiong, M. Yu, X. Guo, S. Chang, W.Y. Wang, Self-supervised learning for contextualized extractive summarization, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2221–2227.
[43] J. Wu, X. Wang, W.Y. Wang, Self-supervised dialogue learning, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 3857–3867.
[44] J. Tang, Z. Lu, J. Su, Y. Ge, L. Song, L. Sun, J. Luo, Progressive self-supervised attention learning for aspect-level sentiment analysis, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 557–566.



Boyuan Pan is currently a Ph.D student in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the B.S. degree in Department of Mathematics from Zhejiang University, China, in 2016. His research interests include machine learning and natural language processing.

Yazheng Yang is currently a master student in the College of Computer Science at Zhejiang University, China. He received the B.S. degree from the School of Computer Science and Technology in Harbin Institute of Technology in 2017. His research interests include machine learning and natural language processing.

Bo Li is an assistant professor in the department of Computer Science at University of Illinois at Urbana-Champaign, and is a recipient of the Symantec Research Labs Fellowship. Prior to this she was a postdoctoral researcher in UC Berkeley. Her research focuses on both theoretical and practical aspects of security, machine learning, privacy, game theory, and adversarial machine learning.

Deng Cai is a Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the PhD degree in computer science from University of Illinois at Urbana Champaign in 2009. Before that, he received his Bachelor's degree and Master's degree from Tsinghua University in 2000 and 2003 respectively, both in automation. His research interests include machine learning, data mining and information retrieval. He is a member of the IEEE.