Hierarchical temporal slot interactions for dialogue state tracking

Junyan Qiu[1,2], Ziqi Lin[2], Haidong Zhang[2*] and Yiping Yang[2]

[1]University of Chinese Academy of Sciences, Beijing, 100190, China.

[2]Institute of Automation, Chinese Academy of Sciences, Zhongguancun East Rd, Beijing, 100190, China.

*Corresponding author(s). E-mail(s): haidong.zhang@ia.ac.cn; Contributing authors: qiujunyan2018@ia.ac.cn; linziqi2013@ia.ac.cn; yiping.yang@ia.ac.cn;

Abstract

Dialogue state tracking (DST), as an essential component of task-oriented dialogue systems, refers to keeping track of the user's intentions as a conversation progresses. Typical methods formulate it as a classification task with fixed pre-defined slot-value pairs, or generate slot-value candidates given the dialogue history. Most of them have limitations on considering interactions of slots with utterance sentences and other slots progressively. To tackle this problem, we propose a Dialogue State Tracker with Hierarchical Temporal Slot Interactions (DST-HTSI) to capture slot-related semantic information from utterance sentences and slots. It firstly captures interactive information among slots within a turn and across turns by applying hierarchical slot interactions. Then a temporal slot interaction module is employed to establish slot dependencies along the time. Finally, a GRU is applied as the decoder to generate values for each slot correspondingly. Furthermore, we also leverage pre-trained language models as the backbone of our model. Experiments show that DST-HTSI outperforms previous state-of-the-art on MultiWOZ 2.2 and WOZ 2.0, and achieves competitive results on MultiWOZ 2.1.

Keywords: Dialogue state tracking, Attention mechanism, Task-oriented dialogue system, Slot interaction.

1 Introduction

Task-oriented dialogue (TOD) systems aim at assisting users in completing specific tasks (e.g., finding restaurants, booking hotel reservation and tickets) in natural language [1]. As its indispensable component, dialogue state tracking (DST) has been attracting growing attentions in both industry and academy communities recently. DST takes as input the dialogue history and outputs of auxiliary components such as Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) [2, 3], then predicts the user's goals or requests of the current turn in the form of slot-value pairs. Each turn may contain more than one state referring to several domains, which complicates the task considerably. For instance, when a user asks *"I need to book a cheap restaurant and a taxi from Cambridge to Stevenage"*, it involves two domains (i.e., *restaurant* and *taxi*) and three states. The system should extract *(restaurant-pricerange, cheap)*, *(taxi-departure, Cambridge)* and *(taxi-destination, Stevenage)* from the utterance.

The rapid progress has motivated an impressive amount of research in DST, which can be mainly categorized into three types: hand-crafted rules, predefined ontology and open vocabulary. The limitation of hand-crafted rules for computing dialogue states is emphasized by the incapability of deriving directly from real dialogue data [2]. Predefined ontology approaches [4–8] are designed as a classification task, selecting the most possible one from a fixed predefined set of values for each slot. Due to their over-dependency on domain ontology, these methods fail to predict the values of free-form slots such as "restaurant-name" and continuous changing value such as "restaurant-book time", which might not be observed in the training data. Both of the laborious collection of slot values and the number of parameters being proportional to the number of slots [9], restrict the generality of predefined ontology methods in real applications.

Open vocabulary methods turn to generate slot values sequentially from the dialogue history without predefined slot values, which can overcome the above challenges. For example, [10, 11] propose a transferable dialogue state tracker which generates values from dialogue history augmented by copy mechanism [12]. [13] selectively overwrites memory with generated values to get more efficient DST.

Nonetheless, the capability of modeling interactions of slots with utterance sentences and other slots is rarely emphasized in previous studies. An illustrated example is shown in Table 1, when a user asks for a taxi to travel between the aforementioned two places at turn 8, the slot *taxi-departure* is related to the system utterance at turn 1 and *taxi-destination* binds to the slot *attraction-name* at turn 4. It is convincible that a model being able to deal with such connections can better track dialogue states in multi-turn dialogues. Hu *et al.* [14] attempts to address the issue by modeling slot interactions using a similarity matrix to control the information flow among similar slots. However, such slot interactions are invariable and independent of the dialogue context. Ouyang *et al.* [15] comes up with an idea that considers the slot correlations

Table 1 An example of multi-domain dialogue state tracking

usr_1:	Find me a modern European restaurant in the south.
states:	(restaurant-food: modern European) (restaurant-area: south)
sys_1:	I found restaurant Alimentum that fits your needs, would you like me to book that for you?
usr_4:	Do you know where Castle Galleries is?
states:	(attraction-name: Castle Galleries)
sys_4:	Yes, the address is unit su43, grand arcade, Saint Andrews street.
usr_8:	I also need a taxi, to go between **the two places**.
states:	(taxi-departure: restaurant Alimentum) (taxi-destination: Castle Galleries)
sys_8:	Sure! when would you like to leave and arrive by?

and predicts the target slot value by directly copying from the source slot at the last turn. Nevertheless, simply concatenating the dialogue history limits its ability to further explore the slot interactions across turns. Moreover, temporal information reflecting the sequence of conversations is neglected, which is crucial when modeling slot interactions. For example, as illustrated in Table 1, the slot *taxi-departure* at turn 8 should consider *attraction-name* at turn 4 when predicting values, otherwise the opposite.

Considering the aforementioned problems, we propose a Dialogue State Tracker with Hierarchical Temporal Slot Interactions, namely DST-HTSI, to capture slot interactions both within a turn and cross turns, and establish slot dependencies along the time. Concretely, it firstly calculates context aggregated representations with a transformer encoder, and slot representations from domain names and slot types. Then we devise hierarchical slot interactions consisting of the local slot interaction and global slot interaction. The local slot interaction obtains slot-specific features by employing a multi-head attention that uses slot representations to guide attention towards context at each turn. And it utilizes another multi-head attention to calculate attention distributions among these slot-specific features to capture inner-turn slot interactions. The global slot interaction calculates multi-head attentions among slot information at current turn and those at previous turns to establish slot correlations across turns. To track the semantic dependencies of the slot representations along the time, we construct a temporal slot interaction module by taking as input the slot representations and cross-turn slot-correlated features. Finally, we apply a GRU as the decoder with slot gate augmented by copy mechanism to generate values for each slot independently in the open vocabulary setting. Furthermore, we also exploit the pre-trained language model BERT as the backbone of DST-HTSI to enhance its language understanding capabilities.

Our work illustrates that progressively capturing slot interactions both within a turn and cross turns hold the promise of advancing the dialogue state tracking performance. We conduct comprehensive experiments on MultiWOZ and WOZ datasets, and DST-HTSI achieves impressive results on them. The ablation study also demonstrates the validity of each module of our model. To conclude, our key contributions are as follows:

- To the best of our knowledge, DST-HTSI is the first to consider slot correlations both within a turn and across turns by modeling slot interactions hierarchically at local and global level.
- DST-HTSI is able to aggregate dependencies of slots along the time by employing temporal slot interaction.
- Our model achieves comparable results with previous state-of-the-art on MultiWOZ 2.1, and outperforms existing predefined ontology and open vocabulary methods by 0.71% and 0.33% on MultiWOZ 2.2 and WOZ 2.0 respectively in terms of joint accuracy.

2 Related work

2.1 Dialogue state tracking

Dialogue state tracking, keeping track of user goals or intents throughout a dialogue, has spurred great interest in the past few years. Traditional dialogue systems use hand-crafted rules to estimate dialogue states by leveraging ASR or NLU outputs [16–18]. Although this method does not require any data to implement, the incapability of benefiting from dialogue data and transferring to new domains severely limit its usage.

To tackle these problems, data-driven approaches have been proposed, which requires little feature engineering and provides strong representation ability. For example, [7] employs BERT [19] to model slots and candidate value representations and scores each slot-value pair in a non-parametric way. [20] leverages various copy mechanisms [21] to extract values from the context on-the-fly for each slot. [22] utilizes graph attention networks to extract schema information and control dialogue state updating. [23] applies large scale pre-training on open domain dialogues and transfers to downstream tasks like DST by performing task-adaptive training. These approaches, however, assuming that all slots and values are predefined, are greatly hindered when the ontology size expands.

Recent studies focus on open vocabulary models in generative fashion. [10] proposes a transferable dialogue state generator to generate values from utterances. [11] further introduces an utterance tagging technique and a bidirectional language model as an auxiliary task to address the problem that the performance drops when the dialogue context sequence is long. To handle unknown slot values, [24] proposes a hierarchical DST framework to identify, update and integrate known and unknown slot values respectively. [13] proposes an efficient dialogue state tracking that selectively generate values depending on the predicted state operation to avoid generating repetitive values. [25] incorporates the prompting technique into DST that provides task-aware context encoding to facilitate slot value generation. To learn more robust DST, [26] proposes a two-pass generation process, where a second pass generation is employed to amend the primitive dialogue state in the first pass.

2.2 Attention mechanism

Attention mechanism, which qualifies the interdependence between two elements or within one element, is firstly introduced to machine translation by [27] and widely applied to other NLP tasks such as dialogue generation [28, 29], natural language understanding [3, 30] and conversational emotion detection [31]. In dialogue state tracking, attention mechanism is usually used to capture the relation between slots and utterance sentence [14, 32] or directly copy words from dialogue history [15]. Recent proposed dialogue state tracker utilizes attention mechanism to model relevance among slots [22, 33]. However, they use only slot names to measure the correlation, making it unchangeable in different contexts, which may result in the correlation being overlooked or overrated. Take the dialogue presented in Table 1 as an example, the slot *taxi-departure* and *taxi-destination* are strongly related in Turn 8. While in other conversations, where one or both of the two slots may not be involved, such relation does not exit. We propose to address such issues by introducing hierarchical slot interactions to model slot correlations according to the context and generate slot values correspondingly.

2.3 Pre-trained language models

Recent advances in pre-trained language models (PLMs) have gained notable performance promotion in DST [7, 20, 23, 26, 34]. PLMs learn general language representations through pre-training on large-scare unstructured corpora with unsupervised learning objectives. They can be generally categorized into three types, i.e., bidirectional, unidirectional and encoder-decoder transformers. Bidirectional transformers like BERT [19], trained to reconstruct the original sequences from the corrupted version, are effective at modeling long texts. It is normally used to extract deep bidirectional semantic features of the dialogue context for obtaining better language understanding capabilities [20, 35, 36]. While unidirectional transformers like GPT-2 [37] learn a distribution for next word prediction based on autoregressive language modeling, which are suitable for sequence generation tasks. For example, [26] exploits GPT-2 [37] and PLATO-2 [38] to amend value generation for DST in two passes. The recent paradigm has evolved to the adoption of encoder-decoder framework [39], which can be viewed as the generalizing of the aforementioned two types of PLMs with both the bidirectional encoder and left-to-right decoder. Specifically, [25] explores the use of sequence-to-sequence pre-trained transformer T5 [39] for prompt-based DST. In this paper, we adopt BERT [19] as the utterance encoder to enhance the understanding capability of our model.

3 Background

In this section, we will briefly introduce the problem statement and two building blocks of this work, namely multi-head attention and transformer.

3.1 Problem statement

DST is responsible for extracting exact slot values pairs from system and user utterances. Formally, a dialogue X of T turns is composed of system utterances S_i and user utterances U_i alternately, i.e., $X = \{D_1, D_2, \cdots, D_T\}$, where each turn utterance D_t is concatenated by one system utterance S_t and one user utterance U_t at the current turn t. Suppose that there are J slots in total $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_J\}$ and each slot $\mathcal{S}_j$ is the combination of a domain name (e.g., *attraction*) and a slot type (e.g., *name*). For a context till turn t, the DST model should assign a value for each slot and predict the dialogue state $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2 \cdots, \mathcal{B}_J\}$, where $\mathcal{B}_j$ is a slot-value pair and only part of them have valid values. Those slots who are not involved in the context will be assigned a meaningless placeholder value like *none*. In the open vocabulary setting, the slot is predefined and the task is to generate the slot value correspondingly given the dialogue history.

3.2 Multi-head attention

The intuition behind multi-head attention (MHA) is that different vectors in a sentence could semantically related to each other in various ways. For instance, given a sentence *"I want a Chinese restaurant to have lunch"*, when computing the representation of the verb *"have"*, it has a great possibility to focus on the noun *"lunch"*. Similarly, it is quite important to attend to the pronoun *"I"* as it is the one who performs the action. Thus [40] proposes a powerful mechanism, allowing us to use different representation sub-spaces of queries $\mathbf{Q}$, keys $\mathbf{K}$ and values $\mathbf{V}$ who are fed into different attention pooling in parallel. Mathematically, MHA$(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ is defined as follows,

$$\begin{aligned}
&\mathbf{Q}^n = \mathbf{Q}\mathbf{W}_Q^n, \mathbf{K}^n = \mathbf{K}\mathbf{W}_K^n, \mathbf{V}^n = \mathbf{V}\mathbf{W}_V^n \\
&\mathbf{head}^n = \frac{\mathbf{Q}^n(\mathbf{K}^n)^T}{\sqrt{d_m}}\mathbf{V}^n \\
&\mathbf{H} = [\mathbf{head}^1, \mathbf{head}^2, \cdots, \mathbf{head}^{n_{head}}]
\end{aligned} \tag{1}$$

where $\mathbf{W}_Q^n, \mathbf{W}_K^n, \mathbf{W}_V^n \in \mathbb{R}^{d_k \times d_k / n_{head}}$ are linear projection matrices, d_m is the model dimensions and n_{head} is the number of heads. $[\cdot, \cdot, \cdots, \cdot]$ denotes the concatenation operation. To make the information flow unidirectionally, i.e., information at current position can only attend to previous positions. The common way is to mask out subsequent vectors and let the vector at current position attend to those at previous positions. Formally, for a matrix $\mathbf{H} = [\mathbf{h}_i, \cdots, \mathbf{h}_k]$ containing k vectors, to calculate the unidirectional multi-head self-attention (UMHA) at position i:

$$\mathrm{UMHA}(\mathbf{h}_i) = \mathrm{MHA}(\mathbf{h}_i, \mathbf{h}_{\leq i}, \mathbf{h}_{\leq i}) \tag{2}$$

3.3 Transformer

Transformer [40] abandons traditional RNN architectures and only uses attention mechanism to process sequential data such as texts and audios. The superiority of transformer over RNN mainly lies in two aspects: 1) Being able to train parallelly in a non-sequential way instead of modeling the sequence word by word, which makes it train faster than RNN with comparable size. 2) Transformer suffers little from long range semantic dependencies as it processes a sentence as a whole and uses positional embeddings to maintain temporal information rather than depending on the historical information. Transformer is composed of an encoder and a decoder. The encoder has two sub-layers. The first sub-layer is a multi-head self-attention mechanism, and the second sub-layer is a feedforward neural network, where a residual connection [41] is employed around each sub-layer followed by a layer normalization (LayerNorm) [42]. In addition to the two sub-layers, transformer decoder added a third sub-layer to perform cross-attention over outputs of the encoder.

4 Methodology

Fig. 1 presents our model consisting of four parts: a) sentence encoder generating utterance and slot representations, b) hierarchical slot interactions modeling slot correlations within a turn and across turns, c) a temporal slot interaction module following each slot's changes along the time and d) a value generator with slot gate generating output tokens for all slots independently.

4.1 Sentence encoder

The sentence encoder includes the utterance encoder and slot encoder. The utterance encoder is mainly to establish semantic dependencies of a turn utterance. In this paper, we use transformer encoder [40] to map all the words $X_t =< sos > \oplus S_t \oplus < sep > \oplus U_t \oplus < eos >$ at turn t to hidden vectors for its superiority of modeling sequences. $\oplus$ means concatenation operation.

$$\mathbf{H}_t^x = \mathrm{Transformer}(\Phi_{emb}(X_t)) \tag{3}$$

where $\mathbf{H}_t^x = [\mathbf{h}_1, \cdots, \mathbf{h}_{|X_t|}]$ is a sequence of vectors. $|X_t|$ is the number of words in the t_{th} turn utterance X_t. Φ_{emb} is the embedding function that maps a token into a fixed-size vector.

For the slot encoder, it takes as input the embedding of the j_{th} domain name $\mathcal{D}_j$ and slot type $\mathcal{R}_j$, then produces the slot encoding by simply adding

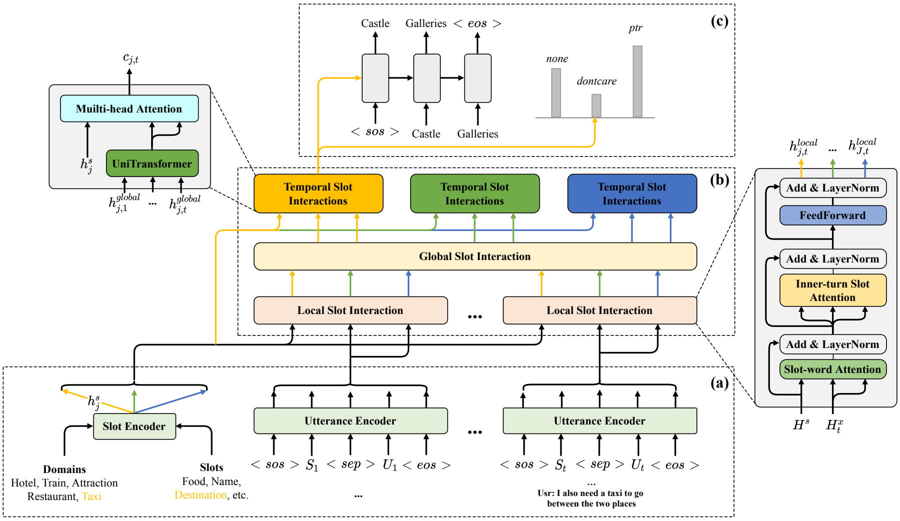


Fig. 1 The architecture of our proposed model, which consists of four components, i.e., (a) sentence encoder, (b) hierarchical slot interactions, (c) temporal slot interaction and (d) slot value generator with slot gate. < *sep* > is a special token that separates the user utterance and system utterance. < *sos* > and < *eos* > represents the start and end of the sequence respectively. Each colored arrow denotes a slot-specific vector flow (e.g., yellow arrow denotes the flow of slot *taxi-destination*). It illustrates an example of the generated value corresponding to the slot *taxi-destination* based on the given utterance at turn t.

them up following [10].

$$\mathbf{h}_j^s = \Phi_{emb}(\mathcal{D}_j) + \Phi_{emb}(\mathcal{R}_j) \tag{4}$$

4.2 Hierarchical slot interactions

In this section, we will introduce the hierarchical slot interactions in two levels, i.e., local and global slot interactions.

4.2.1 Local slot interaction

The local slot interaction module is designed for two purposes: 1) capturing the word-level slot-specific contextual information and 2) modeling slot interactions within a turn. To achieve that, we implement a slot-word attention layer, an inner-turn slot attention layer and a feedforward network layer. Residual connection and layer normalization are used in each layer.

The semantic relevance between slots and turn utterance is obtained by applying a multi-head cross-attention [40] referred as slot-word attention, which takes as input the t_{th} turn word vectors $\mathbf{H}_t^x$ and all slot encoding vectors $\mathbf{H}^s = [\mathbf{h}_1^s, \mathbf{h}_2^s, \cdots, \mathbf{h}_J^s]$, and produces word-level slot-specific vectors

$\mathbf{H}_t^2 = [\mathbf{h}_{1,t}^2, \mathbf{h}_{2,t}^2, \cdots, \mathbf{h}_{J,t}^2]$ for each slot, i.e.,

$$\begin{aligned} \mathbf{H}_t^1 &= \mathrm{MHA}(\mathbf{H}^s, \mathbf{H}_t^x, \mathbf{H}_t^x) \\ \mathbf{H}_t^2 &= \mathrm{LayerNorm}(\mathbf{H}^s + \mathbf{H}_t^1) \end{aligned} \tag{5}$$

Slot interactions within a turn are modeled by another multi-head attention referred as inner-turn slot attention. Formally,

$$\begin{aligned} \mathbf{H}_t^3 &= \mathrm{MHA}(\mathbf{H}_t^2, \mathbf{H}_t^2, \mathbf{H}_t^2) \\ \mathbf{H}_t^4 &= \mathrm{LayerNorm}(\mathbf{H}_t^2 + \mathbf{H}_t^3) \end{aligned} \tag{6}$$

The inner-turn slot attention layer is followed by a feedforward network with Gaussian Error Linear Units (GELU) activation function [43] that produces inner-turn slot-correlated vectors.

$$\begin{aligned} \mathbf{H}_t^{ffn} &= \mathrm{GELU}(\mathbf{H}_t^4 \mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \\ \mathbf{H}_t^{local} &= \mathrm{LayerNorm}(\mathbf{H}_t^4 + \mathbf{H}_t^{ffn}) \end{aligned} \tag{7}$$

4.2.2 Global slot interaction

For the purpose of capturing slot correlations across turns, we devise a global slot interaction module, which enables inner-turn slot-correlated vectors to query those appearing at the current turn or previous turns and produces cross-turn slot-correlated vectors, as shown in Figure. 2. It is worth noting that each vector corresponds to J column vectors, i.e., $\mathbf{H}_t^{local} = [\mathbf{h}_{1,t}^{local}, \mathbf{h}_{2,t}^{local}, \cdots, \mathbf{h}_{J,t}^{local}], t = \{1, 2, \cdots, T\}$, which means there are $J \times T$ vectors in total.

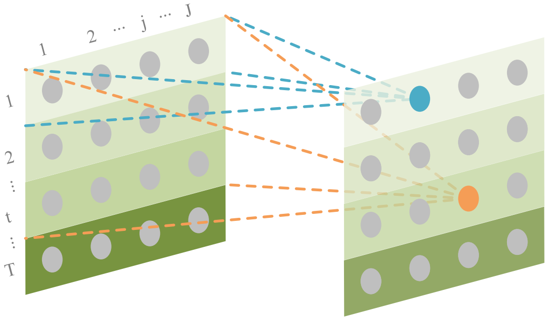


Fig. 2 Illustration of global slot interaction. Circles denote inner-turn slot-correlated vectors. The columns $1 \sim J$ are slot indices, rows $1 \sim T$ are turn indices. Vectors at turn 1 (one example is marked in blue) can only attend to those at turn 1, while vectors at turn t (one example is marked in orange) can attend to those at turn $1 \sim t$.

The global slot interaction module is implemented by a unidirectional transformer encoder with N identical layers. Each layer has two sub-layers. The first sub-layer is a multi-head attention, which is devised to retrieve the relevant slot information. The second sub-layer is a feedforward network (FFN) with GELU activation function. Similar to the local slot interaction, both sub-layers are followed by a residual connection and a layer normalization.

$$\begin{aligned}
\mathbf{h}_{j,t}^{mha} &= \mathrm{MHA}(\mathbf{h}_{j,t}^{n-1}, \mathbf{h}_{1\sim J,1\sim t}^{n-1}, \mathbf{h}_{1\sim J,1\sim t}^{n-1}) \\
\mathbf{h}_{j,t}^{ln} &= \mathrm{LayerNorm}(\mathbf{h}_{j,t}^{mha} + \mathbf{h}_{j,t}^{n-1}) \\
\mathbf{h}_{j,t}^{ffn} &= \mathrm{GELU}(\mathbf{h}_{j,t}^{ln}\mathbf{W}_3^n + \mathbf{b}_3^n)\mathbf{W}_4^n + \mathbf{b}_4^n \\
\mathbf{h}_{j,t}^{n} &= \mathrm{LayerNorm}(\mathbf{h}_{j,t}^{ffn} + \mathbf{h}_{j,t}^{ln})
\end{aligned} \tag{8}$$

where $\mathbf{h}_{j,t}^{n}$ is the output of the n_{th} layer corresponding to slot j at turn t. $\mathbf{h}_{j,t}^{0} = \mathbf{h}_{j,t}^{local}$, $\mathbf{h}_{j,t}^{global} = \mathbf{h}_{j,t}^{N}$.

4.3 Temporal slot interaction

Albeit hierarchical slot interactions are expected to capture possible correlations among slots, the temporal dependencies of the cross-turn slot-correlated vectors corresponding to a specific slot, i.e., $\mathbf{h}_{j,1}^{global}, \mathbf{h}_{j,2}^{global}, \cdots, \mathbf{h}_{j,t}^{global}$, remain to be established. This may result in the model unable to take full advantages of contextual information. Besides, these vectors are supposed to be summarized as one slot-specific context vector to facilitate value generation.

Thus, we propose a temporal slot interaction module with two sub-layers, to address the two issues mentioned above respectively. The first sub-layer is a unidirectional transformer encoder with M identical layers, which is similar to the standard transformer encoder except that the multi-head self-attention is substituted by the unidirectional multi-head self-attention as stated in Equation 2, so that the temporal information can be established. Formally, for the representation of the j_{th} slot at turn t, the temporal information is obtained by[1]:

$$\begin{aligned}
\mathbf{h}_{j,t}^{ctx} &= \mathrm{MHA}(\mathbf{h}_{j,t}^{global}, \mathbf{h}_{j,\leq t}^{global}, \mathbf{h}_{j,\leq t}^{global}) \\
\mathbf{h}_{j,t}^{tmp} &= \mathrm{GELU}(\mathbf{h}_{j,t}^{ctx}\mathbf{W}_1^{tmp} + \mathbf{b}_1^{tmp})\mathbf{W}_2^{tmp} + \mathbf{b}_2^{tmp}
\end{aligned} \tag{9}$$

The second sub-layer is a multi-head attention between the slot encoding and vectors produced by the first sub-layer to obtain the slot-specific context vector.

$$\mathbf{c}_{j,t} = \mathrm{MHA}(\mathbf{h}_j^s, \mathbf{h}_{j,1\sim t}^{tmp}, \mathbf{h}_{j,1\sim t}^{tmp}) \tag{10}$$

[1] Residual connection and layer normalization is omitted for simplicity.

4.4 Slot value generator

After slot interactions, we build the slot value generator using Gated Recurrent Unit (GRU) [44] and generate the dialogue state value sequentially.

$$\begin{aligned}\mathbf{h}^{value}_{i,j,t} &= \text{GRU}(\mathbf{x}_{i,t}, \mathbf{h}^{value}_{i-1,j,t}) \\ \mathbf{p}^{gen}_{i,j,t} &= \text{softmax}(\mathbf{h}^{value}_{i,j,t}\mathbf{W}_{vocab})\end{aligned} \tag{11}$$

where $\mathbf{x}_{i,t}$ is the embedding of the i_{th} word in the target slot value sequence at turn t during training and the last generated word while testing, $\mathbf{h}^{value}_{0,j,t} = \mathbf{c}_{j,t}$. $\mathbf{W}_{vocab}$ is a trainable matrix whose weights are tied with the embedding Φ_{emb} [45] and maps hidden state $\mathbf{h}^{value}_{i,j,t}$ to the vocabulary list.

Aside from computing the distribution probability over the vocabulary, the generator calculates the score for "copying" words from the dialogue history till the current turn.

$$\begin{aligned}\mathbf{p}^{copy}_{i,j,t'} &= \text{softmax}(\mathbf{h}^{value}_{i,j,t}\mathbf{H}_{t'}) \\ \mathbf{p}^{copy}_{i,j,t} &= \sum_{t' \leq t} \mathbf{p}^{copy}_{i,j,t'}\end{aligned} \tag{12}$$

The final distribution is the weighted sum of two distributions and the training objective is defined as the cross entropy between the probability distribution of the generated sequence and true labels $\mathbf{y}_{i,j,t}$.

$$\begin{aligned}\beta &= \text{sigmoid}(\mathbf{h}^{value}_{i,j,t}\mathbf{W}) \in \mathbb{R}^1 \\ \mathbf{p}^{final}_{i,j,t} &= \beta \cdot \mathbf{p}^{gen}_{i,j,t} + (1-\beta) \cdot \mathbf{p}^{copy}_{i,j,t} \\ \mathcal{L}_p &= \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i} -\mathbf{y}_{i,j,t}\log(\mathbf{p}^{final}_{i,j,t})\end{aligned} \tag{13}$$

To reduce the challenge of predicting the slot at current turn, we apply a slot gate following [10]. The slot gate is a three-way classifier to decide the general status in one of $\{none, dontcare, ptr\}$ of the current state, $\mathbf{g}_j = \text{softmax}(\mathbf{c}_j\mathbf{W}_g) \in \mathbb{R}^3$. Only when the *ptr* is predicted will the state value generation be used, otherwise the state value stays the same as the slot gate prediction, i.e., *none* or *dontcare*. The loss of slot gate is defined as the cross entropy between $\mathbf{g}_j$ and the true slot gate label $\mathbf{y}^{gate}_j$.

$$\begin{aligned}\mathcal{L}_{sg} &= \sum_{j} -\mathbf{y}^{gate}_j \log(\mathbf{g}_j) \\ \mathcal{L} &= \mathcal{L}_p + \lambda\mathcal{L}_{sg}\end{aligned} \tag{14}$$

where λ is a hyper-parameter that weights two losses.

5 Experiments and analysis

5.1 Datasets

Table 2 Datasets statistics of MultiWOZ 2.1, 2.2 and WOZ 2.0.

Datasets	**MultiWOZ 2.1/2.2**					**WOZ 2.0**
Domain name ($\mathcal{D}$)	Hotel	Restaurant	Train	Taxi	Attraction	Restaurant
Slot type ($\mathcal{S}$)	area book day book people book stay internet name parking pricerange stars type	area book day book people book time food name pricerange	arriveby book people day departure destination leaveat	arriveby departure destination leaveat	area name type	area food pricerange
Train/Valid/Test	8438/1000/1000					600/200/400

Three datasets MultiWOZ 2.1 [46], MultiWOZ 2.2 [47] and WOZ 2.0 [48] are used to evaluate the validity of our model. MultiWOZ 2.1 is a task-oriented dialogue dataset with 10438 multi-turn dialogues that involves 7 domains in which only five of them (*restaurant, hotel, attraction, taxi, train*) are used following [10]. MultiWOZ 2.2 further fixed the dialogue state annotation errors across 17.3% of the utterances on top of the MultiWOZ 2.1 [47]. WOZ 2.0 is a single domain DST dataset, with 3 slots (*area*, *food* and *price range*) involved in the *restaurant* domain. The dataset statistics are presented in Table 2.

5.2 Training details

The hidden nodes of the utterance encoder, slot interactions and slot value generator are set to 400 following [5, 10, 14, 15]. The number of heads in all transformer encoders (including unidirectional transformer encoder) and multi-head attentions are set to 8. The number of layers of utterance encoder, global slot interaction and unidirectional transformer in temporal slot interaction is 1. Besides, we also employ the base uncased version of BERT [19] with 12 layers of 768 hidden nodes and 12 attention heads as the utterance encoder to enhance the semantic understanding capability of our model. Accordingly, the hidden nodes of subsequent modules, i.e., slot interactions and slot value generator, are set to 768, and the number of heads in slot interactions are set to 12. It is trained using the Adam [49] optimizer with learning rate annealing in the range of [1e-3, 1e-4] for the based model, and with constant learning rate of 1e-5 for the enhanced model. We set the dropout probability to 0.1, batch size to 16. The training procedure stops when the epoch reaches 200 or the validation loss has not fallen for 6 epochs. The teacher forcing rate is set to 0.5 and label smoothing [50] is applied with value 0.1 during training. The loss weight λ in Equation 14 is set to 1.

5.3 Baselines

We carry out comprehensive experiments on the above three datasets by comparing our method with 4 predefined ontology methods and 12 open vocabulary methods.. Among them, SimpleTOD, ConvBERT, TripPy, Seq2Seq-DU, SDP-DST, TripPy+SaCLog, AG-DST and DSGFNet employed pre-trained language models (PLMs)[2] to initialize model parameters. Particularly, ConvBERT and TripPy+SaCLog used extra dialogue datasets to pre-train their modules.

- GLAD [5] uses a global bidirectional LSTM to share parameters between slots, and a local bidirectional LSTM for each slot to learn slot-specific features.
- TRADE [10] generates values for each slot augmented by copy mechanism [21] and slot gate from the concatenated dialogue history.
- DST-SC [15] applies a slot attention to learn slot-specific features from the original context and integrate them using a slot information sharing.
- HDSTM [24] designs a hierarchical framework that derives, updates and integrates the distribution of unknown and known slot values sequentially.
- SAF [51] efficiently utilizes data in a self-supervised manner by introducing an auxiliary pre-training task and an attention flow mechanism to better understand user intents and file out the redundant information.
- DST-picklist [52] proposes a dual-strategy model to jointly handle the categorical and non-categorical slots.
- SOM-DST [13] considers dialogue state as an explicit fixed-sized memory and proposes to selectively overwrite it for more efficient DST.
- SimpleTOD [53] is a generative model for DST that uses a single causal language model trained on three subtasks recast as a sequence prediction problem.
- ConvBERT [23] is a BERT-like model trained on large-scale open domain dialogues to encourage dialogue research in representation-based transfer, domain adaptation, and sample-efficient task learning.
- SDP-DST [25] incorporates the language modeling approach that uses schema-driven prompting into dialogue state tracking.
- SST [22] proposes a schema-guided dialogue state tracker with graph attention networks to predict dialogue states from utterances and schema graphs.
- TripPy [20] fills slots with values copied from the context, predictions from previous turns or system informs.
- Seq2Seq-DU [36] formulates DST as a sequence-to-sequence problem, which leverages two BERTs to model the rich representations of utterances and schemas.
- TripPy+SaCLog [34] combines TripPy with curriculum learning for DST, which requires 337346 dialogue data to pre-train its modules.

[2] In this paper, PLMs particularly refers to pre-trained transformers such as BERT [19] and GPT-2 [37].

- AG-DST [26] devises a two-pass generation process consisting of a basic generation that uses the current turn and previous dialogue state to generate primitive dialogue state, and an amending generation to revise the primitive dialogue state.
- DSGFNet [35] generates a dynamic schema graph to explicitly fuse the prior slot-domain membership relations and dialogue-aware dynamic slot relations.

5.4 Experimental results

Table 3 Experimental results. The column PLMs indicates whether the model utilized pre-trained language models. †, ‡ and ∗ indicate results are borrowed from [25], [26] and [34] respectively.

	Models	**PLMs**	**MultiWOZ**		**WOZ 2.0**
			2.1	**2.2**	
Predefined Ontology	GLAD	✗	-	-	88.10*
	DST-picklist	✗	53.50	-	-
	ConvBERT	✓	58.70	-	93.10*
	SST	✗	55.23	-	-
Open Vocabulary	TRADE	✗	45.60†	48.60†	-
	DST-SC	✗	49.58	-	-
	HDSTM	✗	-	-	84.51
	TriPpy	✓	55.29	-	92.70*
	SAF	✗	51.60	-	-
	SOM-DST	✗	53.68	53.81‡	-
	SimpleTOD	✓	56.45	54.02‡	-
	SDP-DST	✓	56.66	57.60	-
	Seq2Seq-DU	✓	56.10	54.40	91.20
	TriPpy+SaCLog	✓	**60.61**	-	94.20
	AG-DST	✓	-	57.26	91.37
	DSGFNet	✓	56.70	55.80	-
	DST-HTSI	✗	55.81	56.19	92.37
	DST-HTSI$_{bert}$	✓	59.62	**58.31**	**94.53**

We use the joint accuracy as the main evaluation metric to evaluate the performance of the dialogue state tracker, which is defined as the proportion of accurately predicted states at each dialogue turn. It is considered correct if and only if all slot values are correctly predicted.

Table 3 shows the main results of our model and other baselines. The baselines without PLMs include GLAD, DST-picklist, SST, TRADE, DST-SC, HDSTM, SAF and SOM-DST. Compared with these baselines, our proposed DST-HTSI method achieves the best result on all datasets. Among them, the predefined ontology methods include GLAD, DST-picklist and SST, which do not suffer from intrinsic generative defects like generating ill-formatted strings. Compared with methods achieving the best results on MultiWOZ 2.1 and WOZ 2.0 respectively, i.e., SST and GLAD, our DST-HTSI promotes the joint accuracy by 0.58% and 4.27%. The open vocabulary methods without PLMs include TRADE, DST-SC, HDSTM, SAF and SOM-DST. Comparatively, DST-HTSI shows obvious performance promotions, with 1.13% and 2.38% improvement over SOM-DST on MultiWOZ datasets and 7.86% improvement over HDSTM on WOZ 2.0.

Dialogue state trackers that are equipped with PLMs benefit from the rich linguistic knowledge embedded in PLMs [23, 26, 53], and generally gain remarkable performance improvements. To make fair comparisons with these methods, we also employ the base uncased version of BERT [19], the pre-trained deep bidirectional transformer, to initialize the parameters of utterance encoders. The result is presented as DST-HTSI$_{bert}$ in the last row of Table 3. As we can see, DST-HTSI$_{bert}$ indicates more competitive performance, with 3.81%, 2.12% and 2.16% joint accuracy promotion over DST-HTSI on three datasets respectively. Compared with other baselines that are equipped with PLMs, DST-HTSI$_{bert}$ achieves best results among all the baselines on MultiWOZ 2.2 and WOZ 2.0, outperforming the previous state-of-the-art by 0.71% (SDP-DST) and 0.33% (TripPy+SaCLog) respectively. Furthermore, ConvBERT and TripPy+SaCLog utilize a large number extra dialogue data to pre-train their DST models, which generally exhibit more splendid performance than those trained on a single dataset. Nevertheless, our DST-HTSI$_{bert}$ is still comparable with these methods. In fact, it achieves the state-of-the-art on MultiWOZ 2.2 and WOZ 2.0, and is merely exceeded by TripPy+SaCLog with a narrow margin on MultiWOZ 2.1.

5.5 Ablation study

Table 4 Ablation study conducted on MultiWOZ 2.1 and 2.2 datasets measuring the joint accuracy (%).

Model		**MultiWOZ**	
		2.1	**2.2**
DST-HTSI		55.81	55.19
-local slot interactions	-slot word attention	53.13(-2.68)	52.56(-2.63)
	-inner-turn slot attention	54.95(-0.86)	54.38(-0.81)
-global slot interactions		52.13(-3.68)	51.45(-3.74)
-temporal slot interactions		54.16(-1.65)	53.94(-1.25)
-above three		48.87(-6.94)	49.73(-5.46)

To evaluate the impacts of each module influencing the joint accuracy, we conduct several ablation experiments on MultiWOZ datasets. As shown in Table 4, performance degradation can be observed after removing part of the modules, among which casting off the global slot interaction results in the most severe performance drop on MultiWOZ 2.1 and 2.2 by 3.68% and 3.74% respectively. While relatively mild performance decline happens after removing the inner-turn slot attention on these datasets. We argue that this is resulted from the less common slot interactions within a turn comparing with those across turns. As a matter of fact, only 7.90% of slot pairs share the same value within a turn[3] on test dataset of MultiWOZ 2.1. Besides, the global slot interaction consists of attending to slots within a turn, which greatly makes up the absence of inner-turn slot interactions.

To test the effectiveness of temporal slot interaction, we take the mean of cross-turn slot-correlated vectors for the slot-specific context vector after removing it, i.e., Equation 9 is replaced with $\mathbf{c}_j = 1/t \sum_{t'=1}^{t} \mathbf{h}_{j,t'}^{context}$. The result emerges the joint accuracy declines of 1.65% and 1.25% on two datasets respectively, suggesting its capability of modeling the contextual semantic information of cross-turn slot-correlated vectors produced by the global slot interaction.

Furthermore, we implement another model (DST-HTSI$_\Phi$) that discards the above three modules as follows. At turn t, instead of encoding each turn utterance separately, we concatenate utterances till the current turn as a whole sequence and encode them using transformer encoder to obtain context hidden representations $\mathbf{H}_t^x = \text{Transformer}(\Phi_{emb}(X_{\leq t}))$, where $X_{\leq t} = X_1 \oplus \cdots \oplus X_t$. Then the slot-specific context vector is calculated by applying a multi-head attention from the slot encoding $\mathbf{h}_j^s$ to $\mathbf{H}_t^x$, i.e., $\mathbf{c}_{j,t} = \text{MHA}(\mathbf{h}_j^s, \mathbf{H}_t^x, \mathbf{H}_t^x)$. The slot value generation process remains the same as described in Section 4.4. The experimental result is presented in the last row of Table 4. It is obvious that the joint accuracy is much lower than merely removing one module, which demonstrate the crucial impact of hierarchical slot interactions on DST-HTSI. Surprisingly, we notice that it still outperforms TRADE, which shares similar architecture with DST-HTSI$_\Phi$. We argue that this is attributable to the superiority of transformer and multi-head attention over recurrent neural networks and vanilla attention, which are employed by TRADE, in capturing long-range semantic dependencies.

5.6 Slot interactions tests

As stated before, each slot at current dialogue turn may show correlations with those at previous turns. In order to demonstrate the effectiveness of the proposed model dealing with potential slot interactions, we conduct contrast experiments by counting up connected slot pairs and record the joint accuracy achieved by the proposed model with (Model I) and without (Model II) the global slot interaction in Table 5. There are 79 types of connected slot pairs and we listed the top 5 numbers in the table. We find that Model I outperforms Model II on almost every pair and achieves 60.07% overall slot accuracy with 8.91% improvement over Model II. Besides, Table 6 lists the overall accuracy of slots that are not explicitly connected, i.e., slots with different values, and Model I still increases the slot accuracy by 0.85%. It is worth mentioning that

[3] We call such slot pairs as connected pairs for simplicity

Table 5 Slot interactions evaluation on test dataset of MultiWOZ 2.1. † denotes Model I and ‡ denotes Model II, similarly hereinafter

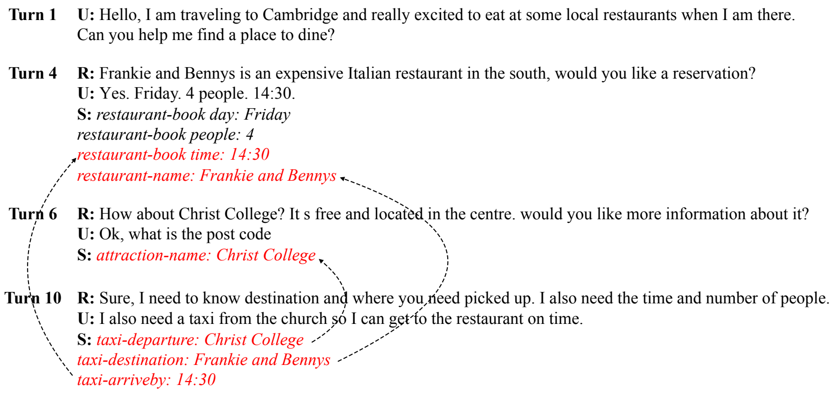
Source Slot	Target Slot	Total	w†	w/o‡
restaurant-book time	taxi-arriveby	57	36	27
hotel-internet	hotel-parking	53	47	39
restaurant-name	taxi-destination	39	27	18
restaurant-area	attraction-area	31	22	19
attraction-name	taxi-departure	26	16	17
Others		703	398	345
Total number		909	546	465
Overall accuracy(%)		-	60.07	51.16

we only count up slots with substantial values and leave out slots with *none* values, resulting in low slot accuracy.

Table 6 Overall slot accuracy of slots that are not connected.

	w	w/o
Overall accuracy(%)	51.18	50.33

5.7 Attention visualization



Turn 1 **U:** Hello, I am traveling to Cambridge and really excited to eat at some local restaurants when I am there. Can you help me find a place to dine?

Turn 4 **R:** Frankie and Bennys is an expensive Italian restaurant in the south, would you like a reservation?
U: Yes. Friday. 4 people. 14:30.
S: *restaurant-book day: Friday*
restaurant-book people: 4
restaurant-book time: 14:30
restaurant-name: Frankie and Bennys

Turn 6 **R:** How about Christ College? It s free and located in the centre. would you like more information about it?
U: Ok, what is the post code
S: *attraction-name: Christ College*

Turn 10 **R:** Sure, I need to know destination and where you need picked up. I also need the time and number of people.
U: I also need a taxi from the church so I can get to the restaurant on time.
S: *taxi-departure: Christ College*
taxi-destination: Frankie and Bennys
taxi-arriveby: 14:30

Fig. 3 Dialogue Example

A dialogue example from test dataset of MultiWOZ 2.1 is presented in Fig. 3 and the corresponding attention visualization of slot word attention and global slot interaction are presented in Fig. 4. We can see from 4(a) that the slot can accurately attend to related words or phrases at turn 4. For instance, the slot *restaurant-book day* attends to *friday* with the highest weights, *restaurant-name* attends to *frankie and bennys* with the highest weights and so on.

In the global slot interaction, we call the slots at turn 10 as target slots and those whose values are where the target slots originate in previous turns as original slots. As can be seen from Fig. 4(b), several intriguing phenomena can be observed: 1) Relations between two identical slots across turns are normally highlighted. For example, the target slot *taxi-destination* put much attention on that at turn 6, so is slot *taxi-departure* and *taxi-arrive by*. 2) Our model is able to capture dependencies between two slots that are semantically related. This can be demonstrated by the attention weight visualization that the target slot *taxi-destination* attends much to the source slot *restaurant-name*, where both of them refer to the same place *Christ College* in the dialogue example. Similar conclusions can be drawn from the attention weight between *taxi-departure* and *taxi-arrive by*. 3) Slots that tend to share the same value are more likely to establish strong interactions. For example, in spite of implicit semantic relations between target slot *taxi-arrive by* and source slot *taxi-leave at*, both of them are normally indicative of time, which makes the interaction stronger than expected. This can also be proved by slot pairs *taxi-destination* at turn 10 and *attraction-name* at turn 6, *taxi-departure* at turn 10 and *hotel-name* at turn 4, all of which are indicative of places.

Furthermore, we also conclude that slot interactions are dependent on the dialogue context, which can be inferred from last two subfigures in Fig. 4(b). Since the contextual information contained in utterances at turn 4 and turn 6 is divergent, attention distributions between the target slot *taxi-arrive by* and other slots at turn 4 and turn 6 exhibit obvious discrepancy. It is attributable to the slot-word attention in the local slot interaction, which incorporates contextual information into slot-specific features and enables to model subsequent slot interactions according to the context.

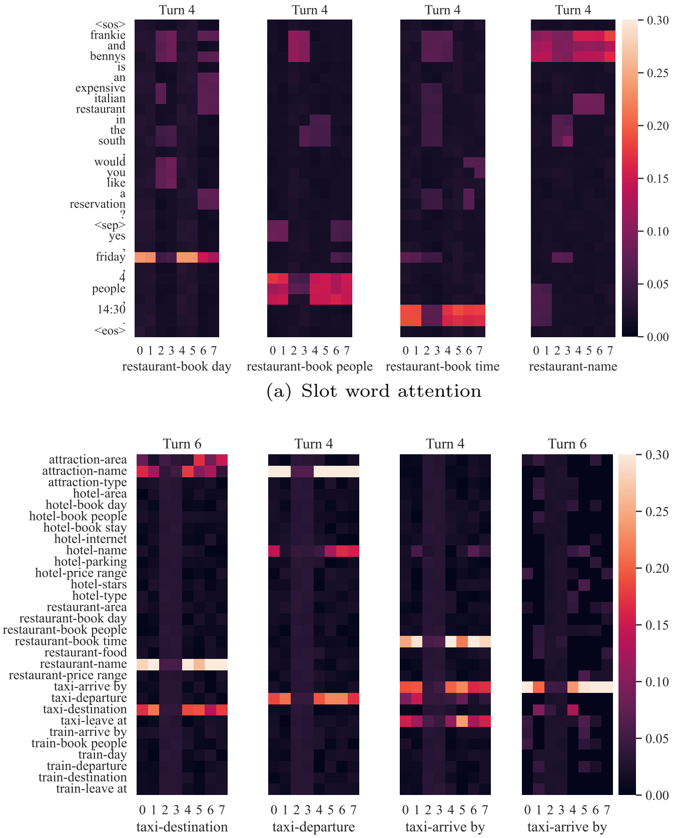
5.8 Discussion and analysis

Per slot accuracy

Since we have proved the validity of slot interaction across turns, we then find out where the improvement come from. To achieve that, we present the accuracy for each slot in Figure 5 on MultiWOZ 2.1 dataset. From the figure we observe that Model I achieves better or comparable results than Model II at most slots. Slight performance declines can be observed at slots with rather small ontology size like *hotel-type* and *hotel-stars*. The results suggest that slot interactions tend to bring accuracy promotion at slots with a large number of possible values, e.g., *restaurant-name*, *hotel-name*, *taxi-departure* and *taxi-destination*.

Influence of the distance between connected pairs

To study the impact of the distance between connected pairs on the performance of Model I and II, we present Fig. 6, which shows the accuracy variation with the difference of turn indices between connected pairs on the test dataset



(a) Slot word attention

(b) Global slot interaction

Fig. 4 Visualizations of slot word attention and global slot interaction on an example from MultiWOZ 2.1 dataset. The columns 0 ∼ 7 denotes the indices of the heads. The padding tokens in Fig. 4(a) are omitted for simplicity.

of MultiWOZ 2.1. In this experiment, we only focus on whether the value of the target slot is correctly predicted. As expected, the performance of Model I outperforms Model II with any difference of turn indices. More specifically, Model II generally exhibits obvious degeneration as the distance between connected pairs becomes longer. On the contrary, for Model I, the slot accuracy does not decline as connected pairs get further. In fact, it achieves the best performance when the difference is 5 with 68.13% accuracy. This can be explained by the fact that slot interactions across turns suffer little from long-range semantic dependencies as multi-head attention obtains the historical information parallelly instead of depending on the sequential information accumulation, which may cause the historical information to be forgotten. However, longer distance between connected pairs normally means longer conversation, resulting in more complex semantic modeling, which can lead to performance degradation, as can be seen from the table when the distance goes to 7 and 8. Moreover, we

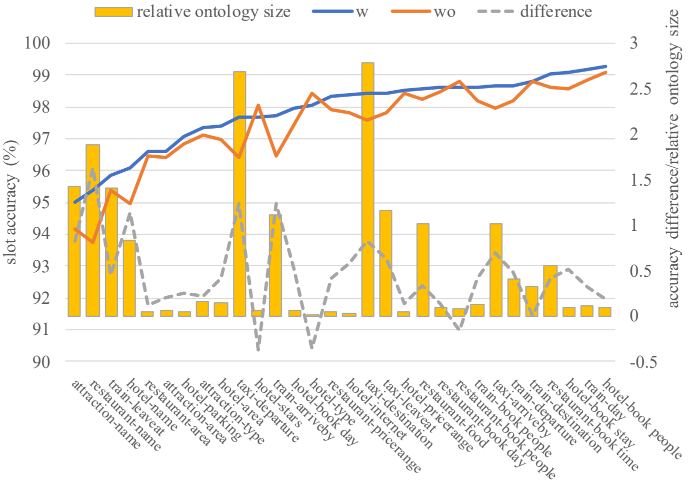
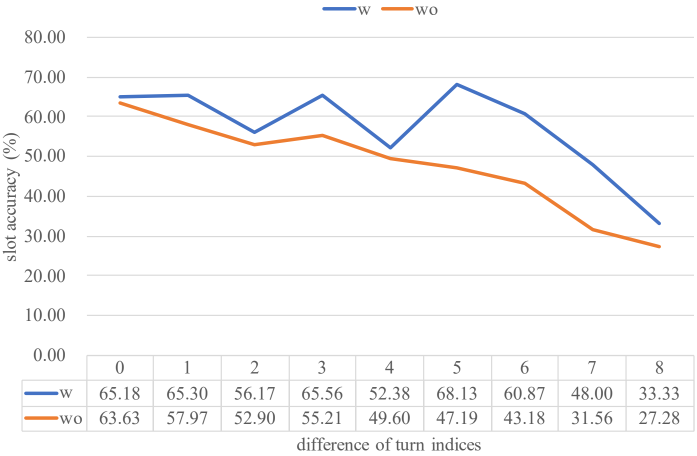


Fig. 5 Slot accuracy on test dataset of MultiWOZ 2.1. The gray dash line denotes the accuracy difference between model I and II. The yellow bar denotes relative ontology size of each slot, which is obtained through dividing the actual value number by 100.



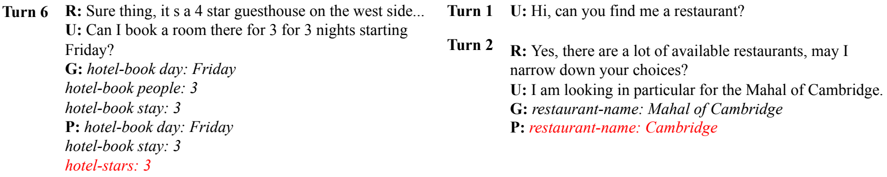
	0	1	2	3	4	5	6	7	8
w	65.18	65.30	56.17	65.56	52.38	68.13	60.87	48.00	33.33
wo	63.63	57.97	52.90	55.21	49.60	47.19	43.18	31.56	27.28

Fig. 6 Accuracy variation with the difference of turn indices on test dataset of MultiWOZ 2.1.

observe that the slot accuracy of Model II is almost the same with Model I when connected pairs are within a turn, i.e., the difference of turn indices is 0. This is because that slot correlations of such connected pairs can be also established by the local slot interaction, which is equivalent to the global slot interaction when modeling slot correlations within a turn.

Error analysis

Typical error examples of generated dialogue states are illustrated in Figure 7. The example on the left presents the type of error that the value of its corresponding slot is implicit, which may lead to the model confusing the correct slot with incorrect ones appearing in the context, i.e., *hotel-stars*. In the second case, the predicted value of slot *restaurant-name* is *Cambridge* while the golden one is *Mahal of Cambridge*. We infer that it is resulted from the more common phrase *Cambridge* appearing in the corpus, which motivates us to do further analysis to diminish this deviation.



Turn 6
R: Sure thing, it s a 4 star guesthouse on the west side...
U: Can I book a room there for 3 for 3 nights starting Friday?
G: *hotel-book day: Friday*
hotel-book people: 3
hotel-book stay: 3
P: *hotel-book day: Friday*
hotel-book stay: 3
hotel-stars: 3

Turn 1
U: Hi, can you find me a restaurant?

Turn 2
R: Yes, there are a lot of available restaurants, may I narrow down your choices?
U: I am looking in particular for the Mahal of Cambridge.
G: *restaurant-name: Mahal of Cambridge*
P: *restaurant-name: Cambridge*

Fig. 7 Two dialogue examples generated by our proposed model. G means golden dialogue states and P means predicted ones. Wrong dialogue states are marked in red.

6 Conclusion and future work

In this paper, we introduce a novel approach DST-HTSI that models slot interactions hierarchically at both local and global level, and establishes temporal slot interactions. In hierarchical slot interactions, DST-HTSI first obtains slot-specific information and inner-turn slot correlations by using local slot interaction. Then a global slot interaction module is leveraged to generate slot-specific vectors for each turn that are semantically related across turns. Moreover, we apply a temporal slot interaction module to capture dependence information of those vectors along the time and summarize them as context vectors to generate values correspondingly. To the end, we employ a pre-trained language model BERT to enhance our model by initializing parameters of the utterance encoder. We conduct experiments on MultiWOZ 2.1, 2.2 and WOZ 2.0, and results demonstrate the validity of our proposed method.

In the future, we will dedicate to solve the problem that out model is apt to generate values that commonly appear in training corpus. While our model is based on the open vocabulary setting, it is applicable to the predefined ontology models. We will explore that and see if this can further improve the performance of our model.

7 Compliance with ethical standards

Conflict of interest We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

8 Data availability

The datasets analyzed during the current study are available in the multiwoz repository (https://github.com/budzianowski/multiwoz for MultiWOZ 2.1 and 2.2 datasets) and N2N-Dialogue-System repository (https://github.com/Yusser95/N2N-Dialogue-System for WOZ 2.0 dataset).

References

[1] Chen, H., Liu, X., Yin, D., Tang, J.: A survey on dialogue systems: Recent advances and new frontiers. Acm Sigkdd Explorations Newsletter **19**(2), 25–35 (2017)

[2] Williams, J.D., Raux, A., Henderson, M.: The dialog state tracking challenge series: A review. Dialogue & Discourse **7**(3), 4–33 (2016)

[3] Ni, P., Li, Y., Li, G., Chang, V.: Natural language understanding approaches based on joint task of intent detection and slot filling for iot voice interaction. Neural Computing & Applications **32**(20) (2020)

[4] Mrkšić, N., Séaghdha, D.Ó., Wen, T.-H., Thomson, B., Young, S.: 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), pp. 1777–1788 (2017)

[5] Zhong, V., Xiong, C., Socher, R.: Global-locally self-attentive dialogue state tracker. arXiv preprint arXiv:1805.09655 (2018)

[6] Nouri, E., Hosseini-Asl, E.: Toward scalable neural dialogue state tracking model. arXiv preprint arXiv:1812.00899 (2018)

[7] Lee, H., Lee, J., Kim, T.-Y.: Sumbt: Slot-utterance matching for universal and scalable belief tracking. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5478–5483 (2019)

[8] Gao, S., Sethi, A., Agarwal, S., Chung, T., Hakkani-Tur, D., AI, A.A.: Dialog state tracking: A neural reading comprehension approach. In: 20th Annual Meeting of the Special Interest Group on Discourse and Dialogue, p. 264 (2019)

[9] Ren, L., Xie, K., Chen, L., Yu, K.: Towards universal dialogue state tracking. arXiv preprint arXiv:1810.09587 (2018)

[10] Wu, C.-S., Madotto, A., Hosseini-Asl, E., Xiong, C., Socher, R., Fung, P.: Transferable multi-domain state generator for task-oriented dialogue systems. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 808–819 (2019)

[11] Quan, J., Xiong, D.: Modeling long context for task-oriented dialogue state generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7119–7124 (2020)

[12] Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1631–1640 (2016)

[13] Kim, S., Yang, S., Kim, G., Lee, S.-W.: Efficient dialogue state tracking by selectively overwriting memory. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 567–582 (2020)

[14] Hu, J., Yang, Y., Chen, C., Yu, Z., *et al.*: Sas: Dialogue state tracking via slot attention and slot information sharing. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6366–6375 (2020)

[15] Ouyang, Y., Chen, M., Dai, X., Zhao, Y., Huang, S., Jiajun, C.: Dialogue state tracking with explicit slot connection modeling. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 34–40 (2020)

[16] Thomson, B., Young, S.: Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. Computer Speech & Language **24**(4), 562–588 (2010)

[17] Wang, Z., Lemon, O.: A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In: Proceedings of the SIGDIAL 2013 Conference, pp. 423–432 (2013)

[18] Henderson, M., Thomson, B., Young, S.: 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), pp. 292–299 (2014)

[19] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[20] Heck, M., van Niekerk, C., Lubis, N., Geishauser, C., Lin, H.-C., Moresi, M., Gasic, M.: Trippy: A triple copy strategy for value independent neural dialog state tracking. In: Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 35–44 (2020)

[21] Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2, pp. 2692–2700 (2015)

[22] Chen, L., Lv, B., Wang, C., Zhu, S., Tan, B., Yu, K.: Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 7521–7528 (2020)

[23] Mehri, S., Eric, M., Hakkani-Tur, D.: Dialoglue: A natural language understanding benchmark for task-oriented dialogue. arXiv e-prints, 2009 (2020)

[24] Yang, G., Wang, X., Yuan, C.: Hierarchical dialog state tracking with unknown slot values. Neural Processing Letters **50**(2), 1611–1625 (2019)

[25] Lee, C.-H., Cheng, H., Ostendorf, M.: Dialogue state tracking with a language model using schema-driven prompting. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 4937–4949 (2021)

[26] Tian, X., Huang, L., Lin, Y., Bao, S., He, H., Yang, Y., Wu, H., Wang, F., Sun, S.: Amendable generation for dialogue state tracking. In: Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, pp. 80–92 (2021)

[27] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

[28] Budzianowski, P., Casanueva, I., Tseng, B., Gasic, M.: Towards end-to-end multi-domain dialogue modelling (2018)

[29] Chen, W., Chen, J., Qin, P., Yan, X., Wang, W.Y.: Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3696–3709 (2019)

[30] Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. arXiv preprint arXiv:1609.01454 (2016)

[31] Ma, H., Wang, J., Qian, L., Lin, H.: Han-regru: hierarchical attention network with residual gated recurrent unit for emotion recognition in conversation. Neural Computing and Applications **33**(7), 2685–2703 (2021)

[32] Kumar, A., Ku, P., Goyal, A., Metallinou, A., Hakkani-Tur, D.: Ma-dst:

Multi-attention-based scalable dialog state tracking. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8107–8114 (2020)

[33] Zhu, S., Li, J., Chen, L., Yu, K.: Efficient context and schema fusion networks for multi-domain dialogue state tracking. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 766–781 (2020)

[34] Dai, Y., Li, H., Li, Y., Sun, J., Huang, F., Si, L., Zhu, X.: Preview, attend and review: Schema-aware curriculum learning for multi-domain dialogue state tracking. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 879–885 (2021)

[35] Feng, Y., Lipani, A., Ye, F., Zhang, Q., Yilmaz, E.: Dynamic schema graph fusion network for multi-domain dialogue state tracking. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 115–126 (2022)

[36] Feng, Y., Wang, Y., Li, H.: A sequence-to-sequence approach to dialogue state tracking. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1714–1725 (2021)

[37] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

[38] Bao, S., He, H., Wang, F., Wu, H., Wang, H., Wu, W., Guo, Z., Liu, Z., Xu, X.: Plato-2: Towards building an open-domain chatbot via curriculum learning. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 2513–2525 (2021)

[39] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**, 1–67 (2020)

[40] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30**, 5998–6008 (2017)

[41] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision

and Pattern Recognition, pp. 770–778 (2016)

[42] Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)

[43] Hendrycks, D., Gimpel, K.: Bridging nonlinearities and stochastic regularizers with gaussian error linear units. arXiv preprint arXiv:1606.08415 (2016)

[44] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS 2014 Workshop on Deep Learning, December 2014 (2014)

[45] Press, O., Wolf, L.: Using the output embedding to improve language models. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 157–163 (2017)

[46] Eric, M., Goel, R., Paul, S., Sethi, A., Agarwal, S., Gao, S., Hakkani-Tür, D.: Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines (2019)

[47] Zang, X., Rastogi, A., Sunkara, S., Gupta, R., Zhang, J., Chen, J.: Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In: Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pp. 109–117 (2020)

[48] Wen, T.-H., Vandyke, D., Mrkšić, N., Gasic, M., Barahona, L.M.R., Su, P.-H., Ultes, S., Young, S.: 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, pp. 438–449 (2017)

[49] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[50] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)

[51] Pan, B., Yang, Y., Li, B., Cai, D.: Self-supervised attention flow for dialogue state tracking. Neurocomputing **440**, 279–286 (2021)

[52] Zhang, J., Hashimoto, K., Wu, C.-S., Wang, Y., Philip, S.Y., Socher, R., Xiong, C.: Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In: Proceedings of the Ninth Joint

Conference on Lexical and Computational Semantics, pp. 154–167 (2020)

[53] Hosseini-Asl, E., McCann, B., Wu, C.-S., Yavuz, S., Socher, R.: A simple language model for task-oriented dialogue. Advances in Neural Information Processing Systems **33**, 20179–20191 (2020)