LUNA: Learning Slot-Turn Alignment for Dialogue State Tracking

Yifan Wang*, **Jing Zhao***, **Junwei Bao**†, **Chaoqun Duan, Youzheng Wu, Xiaodong He**
JD AI Research, Beijing, China
{wangyifan15,zhaojing857,baojunwei,duanchaoqun1,wuyouzheng1,xiaodong.he}@jd.com

Abstract

Dialogue state tracking (DST) aims to predict the current dialogue state given the dialogue history. Existing methods generally exploit the utterances of all dialogue turns to assign value for each slot. This could lead to suboptimal results due to the information introduced from irrelevant utterances in the dialogue history, which may be useless and can even cause confusion. To address this problem, we propose **LUNA**, a **SL**ot-T**U**r**N** **A**lignment enhanced approach. It first explicitly aligns each slot with its most relevant utterance, then further predicts the corresponding value based on this aligned utterance instead of all dialogue utterances. Furthermore, we design a slot ranking auxiliary task to learn the temporal correlation among slots which could facilitate the alignment. Comprehensive experiments are conducted on multi-domain task-oriented dialogue datasets, i.e., MultiWOZ 2.0, MultiWOZ 2.1, and MultiWOZ 2.2. The results show that LUNA achieves new state-of-the-art results on these datasets.[1]

1 Introduction

Dialogue State Tracking (DST) refers to the task of estimating the dialogue state (*i.e.,* user's intents) at every dialogue turn, where the state is represented in forms of a set of slot-value pairs (Williams et al., 2016; Eric et al., 2019). DST is crucial to the success of a task-oriented dialogue system as the dialogue policy relies on the estimated dialogue state to choose actions. Traditional DST approaches assume that all candidate slot-value pairs are predefined in an ontology (Mrkšić et al., 2017; Zhong et al., 2018; Lee et al., 2019). Then, they scores all possible pairs and selecting the value with the highest score as the predicted value of a slot.

Sys:	There are lots to choose from. What type of cuisine are you looking for?
User:	I do not care. It needs to be on the **south** side and moderately priced.
State:	*restaurant-area=south*; *pricerange=moderate*
Sys:	There are 2 options, pizza hut cherry hinton and restaurant alimentum. Can I book you for those ?
User:	Yes please. I also need a hotel with at least 3 stars and free parking near by the restaurant.
State:	*hotel-parking=yes*; *hotel-stars =3*
Sys:	I am sorry, there is no guest house that meets those criteria, either. Would you like to try a different rating, or a different area?
User:	Sure, what about in the city **centre?**
State:	***hotel-area =centre***; *hotel-type=guest house*
⊗:	***hotel-area=south***; *hotel-type=guest house*

Table 1: An example of DST. "**User**" and "**Sys**" means user query and system response respectively. "**State**" is the golden label of dialogue state. "⊗" denotes the predicted states of some existing models and the state marked red is the incorrect prediction.

DST encounters many challenging phenomena unique to dialogue, such as co-references and ellipsis. Consequently, most of existing DST approaches exploit all dialogue utterances in history to assign value for each slot (Shan et al., 2020; Chen et al., 2020a; Quan and Xiong, 2020; Hu et al., 2020; Chen et al., 2020b). However, this could lead to the incorrect value assignment due to the ambiguous contents introduced from some irrelevant utterances with the current slot. As the example shown in Table 1, the models estimate a slot value "*south*" for the slot "*hotel-area*" at turn-3 yet its corresponding golden label is "centre". The reason is that both *"south"* and *"centre"* are the potential slot values to the area-related slot (*i.e.,* "*restaurant-area*" and "*hotel-area*") in the ontology. Actually, the domain of the utterance at turn-1 is "*restaurant*" that is irrelevant to the slot "*hotel-area*".

To address the problem aforementioned, we propose **LUNA**, a **SL**ot-T**U**r**N** **A**lignment enhanced

*Equal contribution.
†Corresponding author: baojunwei001@gmail.com
[1]Our code is available at https://github.com/nlper27149/LUNA-dst

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3319 - 3328
July 10-15, 2022

approach, which divides DST into two sub-tasks: (1) explicitly aligns each slot with its most relevant utterance, (2) assigns the slot value according to the aligned utterance. For example, when predicting the slot value of "*hotel-area*", LUNA first aligns it with the relevant utterance (*i.e.,* turn-3) and then only uses the representations of this utterance to match slot value. Concretely, LUNA consists of four parts: an utterance encoder, a slot encoder, a value encoder, and an alignment module between the first two encoders. The core of LUNA is the alignment module directed at accurate alignment, otherwise there may be a risk of the failure of the second sub-task. Correspondingly, the alignment module equipped in LUNA is performed by an iteratively bi-directional feature fusion network based on the attention mechanism. Some previous works have explored the feature fusion of the two encoders, but they are all uni-directional (Shan et al., 2020; Chen et al., 2020b; Ye et al., 2021), *e.g.,* turn-to-slot or slot-to-turn. Compared with them, the bi-directional way can build a mutual relevance between two encoders and thus more effective for our alignment-oriented objective.

Additionally, we design a ranking-based auxiliary task to supervise LUNA to learn the slot order along with the conversational flow, which could facilitate the alignment. For example, the order of the slots in Table 1 is:

(1) "*restaurant-area*" (2) "*pricerange*" (3) "*hotel-parking*" (4) "*hotel-stars*" (5) "*hotel-area*" (6) "*hotel-type*"

Among the above slots, the most difficult-aligned slot is "*hotel-area*" which confronts the confusion from the utterances at turn-1 (containing "*south*") and turn-3 (containing "*center*"). But the remaining five slots are easy-aligned, such as "*hotel-stars*". If the model combines two information: (1) "*hotel-stars*" is aligned with the utterance at turn-2, (2) the conversation order of "*hotel-area*" is after "*hotel-stars*", it can easily inference that "*hotel-area*" should be aligned with the utterance at turn-3. Notably, our proposed auxiliary task enables LUNA to learn the semantic correlations as well as the temporal correlations among slots. Whereas, existing DST approaches only attempt to model the semantic correlations (Ye et al., 2021; Zhu et al., 2020; Chen et al., 2020b).

Comprehensive experiments are conducted and the results show that LUNA achieves state-of-the-art (SOTA) on three of the most actively studied datasets: MultiWOZ 2.0 (Budzianowski et al., 2018), MultiWOZ 2.1 (Eric et al., 2019), and MultiWOZ 2.2 (Zang et al., 2020) with joint accuracy of 55.31%, 57.62%, and 56.13%. The results outperform the previous SOTA by +0.97%, +1.26%, and +4.43%, respectively. Furthermore, a series of subsequent ablation studies demonstrate the effectiveness of each module in our model. Our main contributions are summarized as follows:

(1) We propose a DST approach LUNA which mitigates the problem of incorrect value assignment through explicitly aligning each slot with its most relevant utterance.

(2) We propose an auxiliary task to facilitate the alignment which is firstly introduced in DST to take the temporal correlations among slots into account.

(3) Empirical experiments are conducted to show that LUNA achieves SOTA results with significant improvements.

2 Related Work

DST is a necessary component in task-oriented dialogue systems and a large amount of work has been proposed to achieve better performance. All these methods can be broadly divided into two categories: classification (Xu and Hu, 2018; Zhong et al., 2018; Ren et al., 2018; Xie et al., 2018) and generation (Wu et al., 2019; Hosseini-Asl et al., 2020; Kim et al., 2020). The classification method requires that all possible slot-value pairs are given in a predefined ontology. Then, the pair with the highest score is the final prediction. Conversely, the generation way does not rely on manual definition, which generates dialogue states from utterances using the seq2seq fashion. This work is mainly related to the classification method.

Recently, transformer-based pre-trained models, such as BERT (Devlin et al., 2019), have achieved remarkable results in a range of natural language processing tasks. Thereupon, the research of DST has been shifted to building new models on top of the powerful pre-trained language models. SUMBT (Lee et al., 2019) is the first model to employ BERT to model the relationships between slots and dialogue utterances through a slot-word attention mechanism. CHAN (Shan et al., 2020) presents a hierarchical attention network which uses slot-word attention and slot-turn attention to enhance the representations of slots. All the methods mentioned above predict the value of each slot separately and ignore the correlations among slots. SST (Chen et al., 2020b) incorporates graph at-

tention networks into DST and proposes schema graphs which contain slot relations in edges. STAR (Ye et al., 2021) provides a slot self-attention mechanism to learn mutual guidance among slots and enhance the ability to deduce appropriate slot values from related slots. Recently, BORT (Sun et al., 2022) proposes a reconstruction mechanism which enhances the performance of DST.

To the best of our knowledge, we are the first to reveal that exploiting all dialogue utterances to assign value may cause suboptimal results and the first to learn the temporal correlations among slots.

3 Methodologies

Suppose that there is a conversation composed of T utterances, $\mathcal{X} = \{(Q_1, R_1), ..., (Q_T, R_T)\}$, and a predefined slot set $\mathcal{S} = \{S_1, ..., S_J\}$, where Q_t denotes the user query at t-th utterance, R_t is the corresponding system response and J is the total number of slots. DST aims to predict states at each turn with given utterances up-to-now $(Q_{\leq t}, R_{\leq t})$, and presents them as slot-value pairs, $\mathcal{B}_t = \{(S_1, V_1^t), ..., (S_J, V_J^t)\}$, where S_j is the j-th slot in $\mathcal{S}$, and V_j^t is the value with respect to S_j for the t-th turn. Since the datasets are collected from multi domains, following previous works (Hu et al., 2020; Kim et al., 2020), we concatenate domain names and slot names as domain specific slots.

To tackle this task, we propose the LUNA model. As depicted in Figure 1, this model consists of three encoders and an alignment network. In this section, we will elaborate each module of this model.

3.1 Encoders

Inspired by the success of the pre-trained model in the community of the NLP, we adopt the BERT (Devlin et al., 2019) to implement the context encoder.

3.1.1 Utterance Encoder

Given the t-th utterance (Q_t, R_t) and its history $(Q_{\leq t}, R_{\leq t})$, we first concatenate them into a single sequence: $\mathcal{X}_t = Q_1 \oplus R_1 \oplus \cdots \oplus Q_t \oplus R_t$. Following the form of the input of the BERT, we then surround the sequence with two special tokens [CLS] and [SEP]. Given that not all of the slots can be aligned to a specific utterance, we further add an extra token [BLANK] as a placeholder. All of the slots that are not mentioned in the dialogue are aligned to [BLANK]. Finally, the input of the utterance encoder can be denoted as follows:

$$X_t = \text{[CLS]} \oplus \mathcal{X}_t \oplus \text{[SEP]} \oplus \text{[BLANK]}. \quad (1)$$

After obtaining X_t, we feed it into the BERT to learn semantic representations:

$$\mathbf{H}_t = \text{BERT}_{finetune}(X_t), \quad (2)$$

where $\mathbf{H}_t = [\mathbf{h}_1^t, \mathbf{h}_2^t, ..., \mathbf{h}_{|X_t|}^t]$, $\mathbf{h}_j^t \in \mathbb{R}^d$. Additionally, we add a learned embedding to every token indicating which turn it belongs to. Thus, for a given token in X_t, its input representation is constructed by summing the corresponding token, position, segment, and turn embeddings. In order to make the BERT more adapt to this task, we fine tune the parameters of the BERT during the training stage.

3.1.2 Slot and Value Encoders

Following previous works (Shan et al., 2020; Ye et al., 2021), we leverage another BERT to encode slots and their candidate values. Formally, given a slot S_j or a value V_j^t, we first tokenize it into a sequence and then concatenate it with the special token [CLS] to build the input for the slot or value encoder. After that, we exploit the BERT to encode the concatenation as follows:

$$\mathbf{h}_{s_j} = \text{BERT}_{fixed}(S_j), \quad (3)$$
$$\mathbf{h}_{v_j^t} = \text{BERT}_{fixed}(V_j^t). \quad (4)$$

We regard the representation of [CLS] as that of the whole slot or value. Specially, since the quantity of the sub-vocabulary related to slots and values are small, we freeze the parameters of BERT in slot and value encoders during the training stage.

3.2 Alignment Module

As mentioned above, DST model usually adopts all of previous utterances as the history to enhance the representation of the current utterance. Although this mechanism enriches the semantic representation, it introduces some noisy and causes confusion for value prediction to a specific slot. To alleviate this issue, the proposed LUNA model adopts iteratively bi-directional feature fusion layers, turn-to-slot and slot-to-turn, to align slots to utterances and provide more relevant utterance for value prediction.

3.2.1 Turn-to-Slot Alignment

In this work, we regard utterances and slots as two sequences and aims to align them with each other. To this end, we first employ a multi-head attention mechanism (Vaswani et al., 2017) to assist

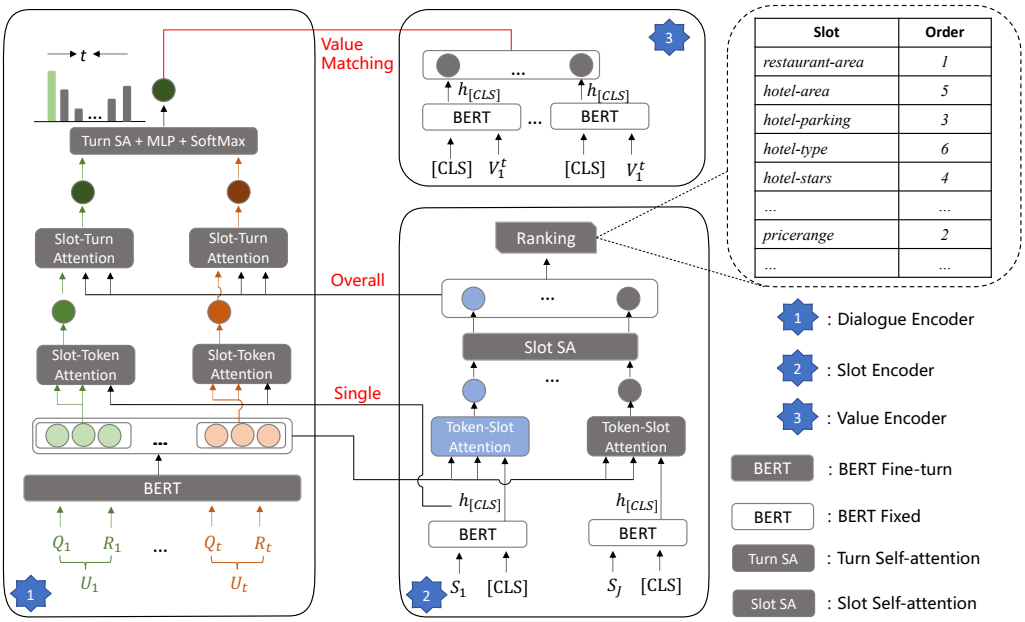

Figure 1: The architecture of LUNA. Note that the workflow in this figure is specifically for the first slot S_1.

the slots extracting relevant information from utterances based on the outputs of the utterance and slot encoders:

$$\mathbf{h}_{s_j,t} = \text{MultiHead}(\mathbf{h}_{s_j}, \mathbf{H}_t, \mathbf{H}_t), \quad (5)$$

where $\text{MultiHead}(\cdot,\cdot,\cdot)$ denotes the multi-head attention mechanism. Through this operation, we obtain utterance-aware slot representations.

After that, we adopt N stacked layers to learn the correlation among slots, and each layer consists of a multi-head self-attention mechanism and a position-wise feed-forward network. We denote this module as Slot SA. Formally, the n-th layer is computed as follows:

$$\bar{\mathbf{H}}_s^n = \text{MultiHead}(\hat{\mathbf{H}}_s^{n-1}, \hat{\mathbf{H}}_s^{n-1}, \hat{\mathbf{H}}_s^{n-1}), \quad (6)$$

$$\hat{\mathbf{H}}_s^n = \text{FNN}(\text{ReLU}(\text{FNN}(\bar{\mathbf{H}}_s^n)) \quad (7)$$

where $\hat{\mathbf{H}}_s^1 = [\mathbf{h}_{s_1,t}, ..., \mathbf{h}_{s_J,t}]$.

3.2.2 Slot-to-Turn Alignment

For utterances, after obtaining the output of the utterance encoder $\mathbf{H}_t$, we first slice it into t segments, $\mathbf{U} = [\mathbf{U}_1, ..., \mathbf{U}_t]$ and each segment corresponds to an utterance. We then exploit a hierarchical attention mechanism to model the slot-to-turn alignment. The hierarchical attention mechanism contains two layers. The first layer models the preliminary alignment between an utterance and a slot and we denote it as **Single Slot-to-Turn**. The other one focuses on the refined alignment through incorporating all slots information and we represent it as **Overall Slot-to-Turn**.

As shown in Figure 1, the Single Slot-to-Turn is responsible for extracting token-level information related to a specific slot from each utterance. Take the j-th slot S_j as an example. Given its representation $\mathbf{h}_{s_j}$, we use it to extract most relevant information from each utterance (*e.g.,* i-th utterance) via the multi-head attention mechanism:

$$\bar{\mathbf{U}}_i = \text{MultiHead}(\mathbf{h}_{s_j}, \mathbf{U}_i, \mathbf{U}_i), \quad (8)$$

where $\bar{\mathbf{U}}_i$ is a d-dimension vector and we regard it as slot S_j aware representation for i-th utterance. Similarly, we obtain slot S_j aware representations for all utterances $\bar{\mathbf{U}} = [\bar{\mathbf{U}}_1, ..., \bar{\mathbf{U}}_t]$ with the same operation.

After that, the Overall Slot-to-Turn layer further aligns utterances with slots. Different with existing work (Ye et al., 2021) of encoding the states of the previous turn B_{t-1} as an information supplement, we first introduce previous alignment information into each utterance by adding alignment embedding:

$$\hat{\mathbf{U}}_i = \bar{\mathbf{U}}_i + \text{AE}(i), \quad (9)$$

where AE is embedding matrix indicating whether the slot S_j aligns utterance $\mathbf{U}_i$ or not at last turn. Then we utilize another multi-head attention module to update utterance representations based on slots information as follows:

$$\tilde{\mathbf{U}}_i = \text{MultiHead}(\hat{\mathbf{U}}_i, \hat{H}_s^N, \hat{H}_s^N). \quad (10)$$

To aggregate context dependency among utterances, we further introduce a multi-head self-attention mechanism to learn the context aware representation for each utterance:

$$\mathbf{D} = \text{MultiHead}(\tilde{\mathbf{U}}, \tilde{\mathbf{U}}, \tilde{\mathbf{U}}), \tag{11}$$

where $\tilde{\mathbf{U}} = [\tilde{\mathbf{U}}_1, ..., \tilde{\mathbf{U}}_t]$, $\mathbf{D} = [\mathbf{D}_1, ..., \mathbf{D}_t]$. $\mathbf{D}$ is adopt to predict the alignment distribution over turns for S_j as follows:

$$p(\cdot|S_j) = \text{softmax}(\mathbf{W}_o\mathbf{D} + b_o), \tag{12}$$

where $\mathbf{W}_o \in \mathbb{R}^d$ and b_o are trainable parameters. We employ the cross-entropy as the objective function of the alignment and it can be formulated as:

$$\mathcal{L}_{align} = -\sum_{j=1}^{J} \log p((Q_j^*, R_j^*)|S_j), \tag{13}$$

where (Q_j^*, R_j^*) is ground-truth utterance aligned to slot S_j.

3.2.3 Auxiliary Ranking Task

The output of the Slot SA, $\hat{\mathbf{H}}_s^N = [\hat{\mathbf{h}}_{s_1}^N, ..., \hat{\mathbf{h}}_{s_J}^N]$, only contains the information of the semantic correlations among slots. To facilitate the alignment, the model needs the assistance of the temporal correlations among slots. However, slots are naturally disordered or sorted in lexicographic order. Therefore, we design an auxiliary task to guide the model to learn the temporal information of slots. Particularly, we propose an ordering algorithm to determine the slots order with respect to the dialogue utterances, as shown in Algorithm 1. This task aims to minimize the order differences between the disordered slots and our defined-ordered slots and we utilize the ListMLE (Xia et al., 2008) as the objective function. ListMLE is a standard ranking loss and it is computed based on a defined list and a ground-truth list. To compute the loss, we learn a score for each slot (*e.g.,* S_j) as follows:

$$\mathbf{f}_{s_j} = \text{Sigmod}(\mathbf{W}_s\hat{\mathbf{h}}_{s_j}^N + b_s), \tag{14}$$

where $\mathbf{W}_s$ and b_s are trainable parameters. Given the ground-truth order of slots $\mathbf{o} = [o_1, ..., o_J]$ and the corresponding slot list is $[S_{o_1}, ..., S_{o_J}]$, the loss function can be formulated as follows:

$$p(j|S_{o_j}) = \frac{\exp(f_{s_{o_j}})}{\sum_{l=j}^{J} \exp(f_{s_{o_l}})}, \tag{15}$$

$$\mathcal{L}_{order} = -\log(\prod_{j=1}^{J} p(j|S_{o_j})). \tag{16}$$

Algorithm 1 Slots Ordering Algorithm

Input: L: Label slots for a conversation, T: the number of turns in this conversation
Initialize: S : A list of sorted slots

```
for t ∈ [1, T] do
    Find the label slots L_t = [l_{t,1}, ..., l_{t,n}] of t-th turn;
    Sort L_t by slots' lexicographic order;
    for l in L_t do
        Add l to S;
    end for
end for
Define the list of remaining not-aligned slots is L_blank;
Sort L_blank by slots' lexicographic order;
for l in L_blank do
    Add l to S;
end for
```

3.3 Value Prediction

Above sections describe the method of aligning slots with utterances. We then predict the value for a specific slot based on the most relevant utterance instead of all of the utterances.

Formally, given a slot S_j, we first select the most relevant utterance (Q_j^*, R_j^*) as follows:

$$(Q_j^*, R_j^*) = \arg\max(\{p((Q_i, R_i)|S_j)\}_{i=1}^{t}). \tag{17}$$

Then we feed its representation $D^* \in D$ into a linear layer which is followed by a layer normalization:

$$O^* = \text{LayerNorm}(\text{Linear}(D^*)). \tag{18}$$

Following Ren et al. (2018), we adopt the L2-norm to compute the distance between a slot and a candidate value. Thereby, the value prediction probability distribution can be formulated as follows:

$$p(V_j^t|(Q_{\leq t}, R_{\leq t}), S_j) = \frac{\exp(-\|O^* - h_{v_j^t}\|_2)}{\sum_{V_k^t \in V^t} \exp(-\|O^* - h_{v_k}^t\|_2)}, \tag{19}$$

where V^t is the set of candidate value of slot S_j for the t-th utterance. Finally, the loss function can be defined as:

$$\mathcal{L}_{value} = -\sum_{j=1}^{J} \log(p(V_j^t|(Q_{\leq t}, R_{\leq t}), S_j)). \tag{20}$$

Model	MultiWOZ 2.0		MultiWOZ 2.1		MultiWOZ 2.2		Trainable Parameters
	Joint	Slot	Joint	Slot	Joint	Slot	
Generation models							
SOM-DST (Kim et al., 2020)	51.38	-	52.57	-	-	-	113M
TRADE (Wu et al., 2019)	48.60	96.92	45.60	-	45.40†	-	-
TripPy (Heck et al., 2020)	53.51	-	55.32	-	53.52	-	110M
TripPy w/o LM	45.64†	-	44.80†	-	-	-	-
Seq2Seq-DU (Feng et al., 2021)	-	-	56.10	-	54.40	-	220M
SimpleTOD (Hosseini-Asl et al., 2020)	51.37	-	51.89	-	-	-	-
Classification models							
DS-DST (Zhang et al., 2020)	-	-	51.31	97.35	51.70	-	-
DST-Picklist (Zhang et al., 2020)	54.39	-	53.30	97.40	-	-	-
CHAN (Shan et al., 2020)	53.06	-	53.38	-	-	-	133M
SST (Chen et al., 2020b)	51.17	-	55.23	-	-	-	-
STAR (Ye et al., 2021)	54.34	-	56.36	97.51	-	-	135M
LUNA	**55.31**	**97.35**	**57.62**	**97.96**	**56.13**	**97.68**	142M
With Data Augmentation							
TripPy+ConvBERT (Mehri et al., 2020)	-	-	58.70	-	-	-	-
TripPy+CoCoAug (Li et al., 2020)	-	-	60.53	-	-	-	-
TripPy+SaCLog (Dai et al., 2021)	-	-	60.61	-	-	-	-

Table 2: Joint accuracy (%) and slot accuracy (%) on the test sets. "LM" denotes label map in TripPy. † indicates the reproduced results using the source codes and remaining results reported in the literature.

3.4 Optimization

We adopt the multi-task learning to jointly optimize the alignment loss, value prediction loss and the auxiliary task loss. The total loss is defined as follows:

$$\mathcal{L}_{joint} = \mathcal{L}_{order} + \mathcal{L}_{align} + \mathcal{L}_{value}$$

4 Experimental Setup

4.1 Datasets and Metrics

We evaluate our approach on three gradually refined task-oriented dialogue datasets: MultiWOZ 2.0 (Budzianowski et al., 2018), MultiWOZ 2.1 (Eric et al., 2019), and the latest MultiWOZ 2.2 (Zang et al., 2020), containing over 10,000 dialogues, 7 domains, and 35 domain-slot pairs. MultiWOZ 2.1 modifies about 32% of the state annotations in MultiWOZ 2.0. MultiWOZ 2.2 is the latest and a further refined version of MultiWOZ 2.1, which solves the inconsistency of state updates and some problems of ontology.

We use joint accuracy and slot accuracy as our evaluation metrics. Joint accuracy is the proportion of dialogue turns where the value of each slot is correctly predicted. Slot accuracy only considers individual slot-level accuracy. The ground-truth of slot value is set to none if the slot has not been mentioned in dialogue.

4.2 Training

Same as the previous work (Shan et al., 2020; Ye et al., 2021), we use BERT-base-uncased model as the encoders of LUNA where only the utterance encoder is fine-tuned and the parameters of the other two encoders are fixed. BERT-base has 12 layers of 784 hidden units and 12 self-attention heads. The number of attention heads in multi-head attention in our alignment module is set to 4. The number of layers in slot self-attention and turn self-attention is set to 4 and 2 respectively. During the training process, we use Adam optimizer (Kingma and Ba, 2015) and set the warmup proportion to 0.1. Considering that the encoder is a pre-trained BERT model while the other parts in our model needs to be trained from scratch, we use different learning rates for those parts. Specifically, the peak learning rate is set to 3e-5 for the utterance encoder and 1e-4 for the remaining parts. The maximum input sequence length in BERT is set to 512. For MultiWOZ 2.0, MultiWOZ 2.1, and MultiWOZ 2.2, we apply the same hyperparameter settings.

5 Experiment Results

5.1 Main Results

Table 2 shows the joint accuracy and the slot accuracy of our model and other baselines on the test sets of MultiWOZ 2.0, 2.1, and 2.2, where some models are not tested on the 2.2 version since it was released shortly. As shown in the table, among the models without data augmentation, our model LUNA achieves state-of-the-art performance on these datasets with joint accuracy of 55.31%, 57.62%, and 56.13%, which has a measurable im-

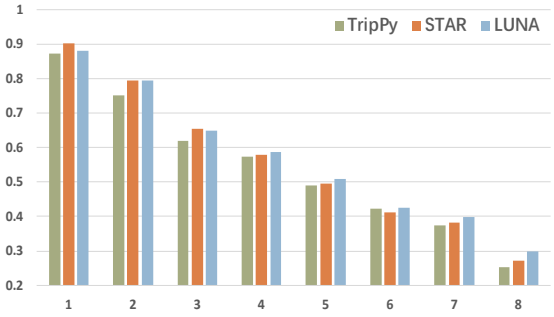

Figure 2: Joint accuracy at every turn.

provement (0.97%, 1.26%, and 4.43%) over the previous best results, illustrating the effectiveness of slot-turn alignment in DST task.

It can be observed that the three data augmented methods reach higher than 58% joint accuracy on MultiWOZ 2.1. We believe that these data augmentation skills are versatile. If they can improve the results of TripPy that lags behind our model, we reasonably speculate that these skills can also improve the effect of LUNA. Besides, all these models are based upon TripPy, which employs a label map as extra supervision. The label map is a dictionary of synonyms, which is used during the testing phase. For example, the official label of the slot *"hotel-area"* in ontology is *"centre"*. But label map regards all its synonyms, such as *"center"*, are also the ground truth. We think that this manner severely reduces the difficulty of the DST task. As shown in Table 2, the performance of TripPy degrades dramatically when the label map is removed. By contrast, our model does not rely on any extra information and is more generalized.

Additionally, Table 2 lists the number of trainable parameters of some baselines and our model, which illustrates that our alignment module containing multiple self-attention does not introduce large model parameters. Compared with the baselines, the size of our model is comparable.

Accuracy at Every Turn. In practice, the dialogue states of longer dialogues tend to be more difficult to be correctly predicted as the model needs to consider more dialogue history. In this section, we further analyze the relationship between the depth of conversation and the prediction accuracy. The joint accuracy at every turn of TripPy, STAR, and LUNA on MultiWOZ 2.1 test set is shown in Figure 2. It presents that the scores of LUNA and STAR are basically the same when the number of conversation turns is less than 3. While as the conversation turns increases from 3, the superiority of LUNA gradually becomes obvious. This is because that both TripPy and STAR exploit all dialogue utterances to assign value for each slot. This may introduce more useless information that causes confusion to the current slots. Whereas, LUNA only uses the most relevant utterance to assign slot value, which avoids interference by useless information.

5.2 Ablation on Alignment Module

To explore the effectiveness of each part in our proposed alignment module, we conduct an ablation study of these parts on the test set of MultiWOZ 2.1, as shown in Table 3.

Model	Align Acc	Joint Acc.
LUNA	97.50	57.62
- Alignment module	–	53.46 (-4.16)
- Overall slot-to-turn alignment	95.23 (-2.27)	54.70 (-2.92)
- Auxiliary task	96.30 (-1.20)	55.29 (-2.33)

Table 3: The ablation study of the alignment module on the MultiWOZ 2.1. Alignment accuracy (%) is defined as the ratio of dialogue for which the utterance turn of each slot is correctly aligned.

First, we remove the whole alignment module and only use the representations of slots obtained by token-slot attention Eq.5 to match the value. The results show that model performance has dropped a lot (4.16 joint accuracy), proving that there are many useless tokens in the conversation history, which interfere the prediction accuracy of slot value. Next, we remove the layer of overall slot-to-turn alignment. We can see that this also severely damages the model performance on both alignment accuracy and joint accuracy. This illustrates that it is not enough to only use the information of a single slot for the alignment. The model needs to comprehensively consider all slots information, such as semantic correlations and temporal correlations among slots to accurately align slots and dialogue turns. Finally, we remove the auxiliary ranking task and the results decrease by 1.20 on alignment accuracy and 2.33 on joint accuracy. This proves that the temporal correlation among slots is important in our model which could facilitate the alignment, as we explained in the section of Introduction. More intuitive explanations will be given in the next section through an example of visualization.

Hard or Soft Alignment. From Table 3, although our hard alignment is highly accurate (Acc 97.50), we should further explore whether it can be replaced by a soft alignment to avoid the risk of error propagation. Whereupon, we design a soft align-

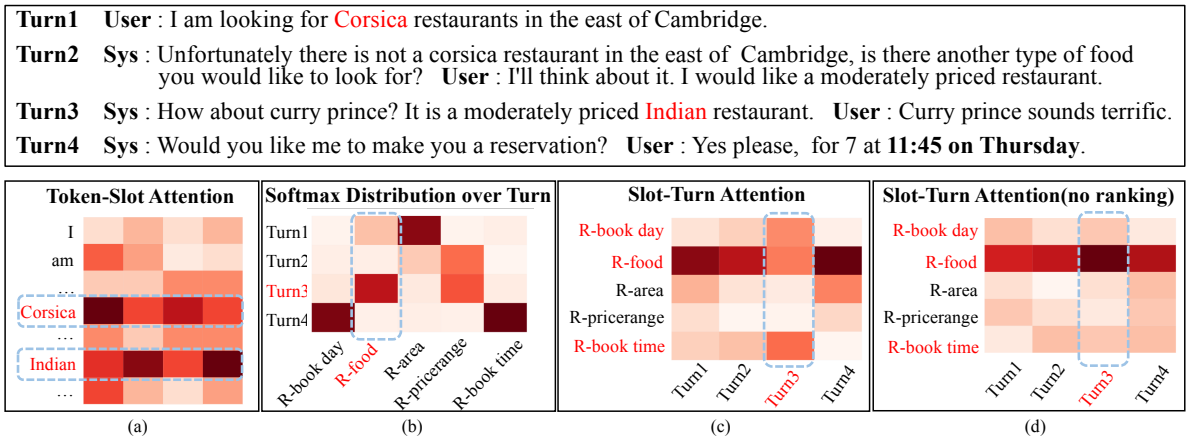

Figure 3: Visualization of LUNA on an example from MultiWOZ 2.1, which is a process of predicting the value to slot "restaurant-food" at turn-4. "R-" in figure denotes "restaurant-". The golden value of "restaurant-food" is "Indian" and the confusion value is "Corsica". (a) is the distribution of Token-Slot Attention calculated by Eq.5 where the columns are the four heads in multi-head attention. (b) is the softmax distribution of alignment over turns calculated by Eq.12. (c)(d) are the distributions of Slot-Turn Attention calculated by Eq.10 where (d) is the version after removing ranking loss.

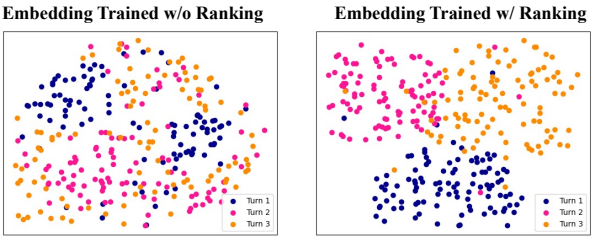

Figure 4: Visualization of slot embedding using t-sne. Each point represents the slot aligned to the corresponding turn. We plot 100 slot embeddings for each turn.

ment that is a weighted sum of all turns with the alignment distribution over the turns as weights (Eq. 12). The experimental results show that compared with hard alignment, the Joint accuracy of soft alignment on MultiWOZ 2.1 drops to 57.53 (-0.09). The reason is that the soft alignment encounters the problem of noises introduced from irrelevant utterances. In other words, risk of error propagation and noise-avoiding are a trade-off. The experimental results show that the benefits of our proposed hard alignment outweigh the risk.

5.3 Visualization

Figure 3 gives an example to visualize the process of predicting the value to slot "restaurant-food". In this example, the golden slot value is "Indian" and "Corsica" is the confusion value. As shown in sub-figure (a), the slot assigns high attention weight in all heads to both "Indian" and "Corsica" , because as of this step, it cannot determine which one is the correct value. At the last step, after the bi-directional fusion in our proposed alignment module, the model successfully assigns turn-3 a larger alignment score than turn-1, as shown in sub-figure (b). In other words, the model has realized that the utterance of turn-3 (containing "Indian") is more important than turn-1 (containing "Corsica"). This can avoid the confusion caused by "Corsica".

We next analyze sub-figures (c) and (d). As we can see, all turns focus on the slot "restaurant-food" as they incorporates its single slot information through Eq.8. For the column of turn-3, if the model is supervised with the auxiliary ranking task, it will also consider the information of "restaurant-book day" and "restaurant-book time". Sub-figure (b) indicates that these two slots are easy-aligned (with turn-4). Meanwhile, the model learns that the order of the three slots is ["restaurant-food", "restaurant-book day", "restaurant-book time"]. Thereby, the alignment of "restaurant-food" and turn-3 becomes easier.

Figure 4 displays the 2-d visualization of slot embeddings obtained by Eq. 7. It can be seen that without ranking loss, the slot representations are irregular and borderless. Under the supervision of the ranking loss, the model can learn the boundaries between the slots aligned with different turns.

6 Conclusion

In this work, we reveal the problem in DST that exploiting all dialogue utterances to assign value to slots may cause suboptimal results. To alleviate it, we propose LUNA, a slot-turn alignment

enhanced approach. and design a ranking-based auxiliary task to supervise LUNA to learn the temporal correlations among slots. Comprehensive experiments are conducted on MultiWOZ 2.0, 2.1, and 2.2 and the results show that LUNA achieves new state-of-the-art results. Moreover, the visualization demonstrates the interpretability of LUNA.

7 Acknowledge

We would like to thank the anonymous reviewers for their useful feedback. This work is supported by the National Key Research and Development Program of China under Grant No. 2020AAA0108600.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. 2020a. Parallel interactive networks for multi-domain dialogue state generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1921–1931, Online. Association for Computational Linguistics.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020b. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7521–7528. AAAI Press.

Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, and Xiaodan Zhu. 2021. Preview, attend and review: Schema-aware curriculum learning for multi-domain dialog state tracking. *ArXiv preprint*, abs/2106.00291.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tür. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines.

Yue Feng, Yang Wang, and Hang Li. 2021. 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)*, pages 1714–1725, Online. Association for Computational Linguistics.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. 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*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jiaying Hu, Yan Yang, Chencai Chen, Liang He, and Zhou Yu. 2020. SAS: Dialogue state tracking via slot attention and slot information sharing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6366–6375, Online. Association for Computational Linguistics.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2020. Coco: Controllable counterfactuals for evaluating dialogue state trackers. *ArXiv preprint*, abs/2010.12850.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *ArXiv preprint*, abs/2009.13570.

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

Jun Quan and Deyi Xiong. 2020. Modeling long context for task-oriented dialogue state generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7119–7124, Online. Association for Computational Linguistics.

Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. Towards universal dialogue state tracking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2786, Brussels, Belgium. Association for Computational Linguistics.

Yong Shan, Zekang Li, Jinchao Zhang, Fandong Meng, Yang Feng, Cheng Niu, and Jie Zhou. 2020. A contextual hierarchical attention network with adaptive objective for dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6322–6333, Online. Association for Computational Linguistics.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. Bort: Back and denoising reconstruction for end-to-end task-oriented dialog. In *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Jason D Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 1192–1199. ACM.

Kaige Xie, Cheng Chang, Liliang Ren, Lu Chen, and Kai Yu. 2018. Cost-sensitive active learning for dialogue state tracking. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 209–213, Melbourne, Australia. Association for Computational Linguistics.

Puyang Xu and Qi Hu. 2018. 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)*, pages 1448–1457, Melbourne, Australia. Association for Computational Linguistics.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, pages 1598–1608.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. 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*, pages 109–117, Online. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip Yu, Richard Socher, and Caiming Xiong. 2020. 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*, pages 154–167, Barcelona, Spain (Online). Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.

Su Zhu, Jieyu Li, Lu Chen, and Kai Yu. 2020. Efficient context and schema fusion networks for multi-domain dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 766–781, Online. Association for Computational Linguistics.