A Contextual Hierarchical Attention Network with Adaptive Objective for Dialogue State Tracking

Yong Shan[1,2]†**, Zekang Li**[1,2]**, Jinchao Zhang**[3]**, Fandong Meng**[3]**, Yang Feng**[1,2]*
Cheng Niu[3]**, Jie Zhou**[3]
[1] Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2] University of Chinese Academy of Sciences
[3] Pattern Recognition Center, WeChat AI, Tencent Inc, China
{shanyong18s,lizekang19g,fengyang}@ict.ac.cn
{dayerzhang,fandongmeng,chengniu,withtomzhou}@tencent.com

Abstract

Recent studies in dialogue state tracking (DST) leverage historical information to determine states which are generally represented as slot-value pairs. However, most of them have limitations to efficiently exploit relevant context due to the lack of a powerful mechanism for modeling interactions between the slot and the dialogue history. Besides, existing methods usually ignore the slot imbalance problem and treat all slots indiscriminately, which limits the learning of hard slots and eventually hurts overall performance. In this paper, we propose to enhance the DST through employing a contextual hierarchical attention network to not only discern relevant information at both word level and turn level but also learn contextual representations. We further propose an adaptive objective to alleviate the slot imbalance problem by dynamically adjust weights of different slots during training. Experimental results show that our approach reaches 52.68% and 58.55% joint accuracy on MultiWOZ 2.0 and MultiWOZ 2.1 datasets respectively and achieves new state-of-the-art performance with considerable improvements (+1.24% and +5.98%). [1]

1 Introduction

Recently, task-oriented dialogue systems have attracted increasing attention in both industry and academia due to their broad application for helping users accomplish tasks through spoken interactions (Young, 2002; Young et al., 2013; Gao et al., 2019a). Dialogue state tracking (DST) is an essential part of dialogue management in task-oriented dialogue systems. Given current utterances and dialogue history, DST aims to determine the set of goals that a user tries to inform at each turn which are represented as slot-value pairs (Williams et al., 2013; Henderson et al., 2014a).

User: Hello, I'm looking for a resraurant with fair prices. **State:** *price range=moderate*
Sys: OK. There are Golden Wok Chinese restaurant and Nirala which serves Indian food, which one do you like? **User:** Are they both have a reasonable price? **State:** *price range=moderate*
Sys: Of course. **User:** Please tell me the address of Golden Wok. **State:** *price range=moderate; food=chinese*

Table 1: An example dialogue. At the last turn, it is necessary to capture relevant information in dialogue history to correctly predict the value of slot "*food*", which is underlined. "User" and "Sys" represent user utterance and system response respectively, and the italic text means dialogue states.

As Table 1 shows, the dialogue state is usually dependent on relevant context in the dialogue history, which is proven in previous studies (Sharma et al., 2019; Wu et al., 2019). However, traditional DST models usually determine dialogue states by considering only utterances at current turn (Henderson et al., 2014b; Mrkšić et al., 2017; Zhong et al., 2018; Chao and Lane, 2019) which neglects the use of dialogue history. Recent researches attempt to address this problem through introducing historical dialogue information into the prediction of slot-value pairs. Most of them leverage a naive attention between slots and concatenated historical utterances (Wu et al., 2019; Zhou and Small, 2019; Gao et al., 2019b; Zhang et al., 2019; Le et al., 2020a,b) or only utilize partial history (Ren et al., 2019; Kim et al., 2019; Sharma et al., 2019) or lack direct interactions between slots and history (Ren et al., 2018; Lee et al., 2019; Goel et al., 2019). Briefly, these methods are deficient in exploiting relevant context from dialogue history.

† Joint work with Pattern Recognition Center, WeChat AI, Tencent Inc.
* Yang Feng is the corresponding author.
[1] Code is available at https://github.com/ictnlp/CHAN-DST

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6322–6333
July 5 - 10, 2020.

Furthermore, there are differences in the frequency of different slots and different slot-value pairs. For example, in MultiWOZ 2.0 train set, there are 15384 samples related to the slot "*train-day*" while 5843 for the slot "*attraction-name*"; the slot-value pair (*attraction-area*, *center*) occurs 5432 times and (*taxi-departure*, *royal spice*) occurs only 9 times; etc. We refer to this problem as "slot imbalance", which makes the learning difficulties of different slots varies (Refer to Appendix for details). However, existing approaches usually ignore the slot imbalance problem and treat all slots indiscriminately, which limits the learning of those hard slots and eventually damages the overall performance.

To address the two aforementioned problems, we propose an effective model equipped with a **c**ontextual **h**ierarchical **a**ttention **n**etwork (CHAN) to fully exploit relevant context from dialogue history, and an **adaptive objective** to alleviate the slot imbalance problem. In CHAN, the slot firstly retrieves word-level relevant information from utterances at each turn. Then, these word-level relevant information will be encoded into contextual representations by rich interactions. Finally, the slot aggregates all contextual representations into turn-level relevant information and then we combine it with word-level relevant information to obtain the outputs. To further enhance the ability to exploit relevant context, we employ a state transition prediction task to assist DST learning. For the slot imbalance problem, our adaptive objective can dynamically evaluate the difficulties in an accuracy-sensitive manner and then adaptively adjust the learning weights for different slots. Thus, it can balance the learning of all slots as far as possible.

We evaluate the effectiveness of our model on MultiWOZ 2.0 and MultiWOZ 2.1 datasets. Experimental results show that our model reaches 52.68% and 58.55% joint accuracy, outperforming previous state-of-the-art by **+1.24%** and **+5.98%**, respectively. The ablation study also demonstrates each module's effectiveness in our model. Our contributions are as follows:

- We propose an effective contextual hierarchical attention network to fully exploit relevant context from dialogue history and employ a state transition prediction task to further enhance it.

- We design an adaptive objective to address the slot imbalance problem by dynamically adjusting the weight of each slot. To the best of our knowledge, our method is the first to address the slot imbalance problem in DST.

- Experimental results show that our model achieves state-of-the-art performance with significant improvements over all previous models.

2 Approach

As shown in Figure 1, the proposed model consists of three components: 1) the **c**ontextual **h**ierarchical **a**ttention **n**etwork (CHAN); 2) the state transition prediction module; 3) the adaptive objective. We share all the model parameters for each slot to keep our model universal for all slots.

2.1 Problem Statement

Given a dialogue $X = \{(U_1, R_1), \ldots, (U_T, R_T)\}$ of T turns where U_t represents user utterance and R_t represents system response of turn t, we define the dialogue state for each turn t as $\mathcal{B}_t = \{(s, v_t), s \in \mathcal{S}\}$ where $\mathcal{S}$ is a set of slots and v_t is the corresponding value of the slot s. Following Lee et al. (2019), we use the term "slot" to refer to the concatenation of a domain name and a slot name in order to represent both domain and slot information. For example, "*restaurant-food*". Similar to (Ren et al., 2018; Lee et al., 2019), we decompose the dialogue state tracking to a multi-label classification problem where we score each value with slot-related features in a non-parametric way and then choose the best candidate. We also add a literally "*none*" into the value set of each slot to represent that no corresponding value is tracked.

2.2 Contextual Hierarchical Attention Network

Recently the pre-trained BERT language model (Devlin et al., 2019) shows powerful ability in universal contextual semantics representation, thus we employ BERT to encode utterances, slots and values. To better retrieve relevant context from dialogue history, we devise Slot-Word Attention and Slot-Turn Attention to query both relevant keywords and turns. Specifically, we exploit a Context Encoder between word-level and turn-level attention to capture contextual representations of relevant information from dialogue history. Furthermore, we devise a Global-Local Fusion Gate to balance the information from global context and local utterances.

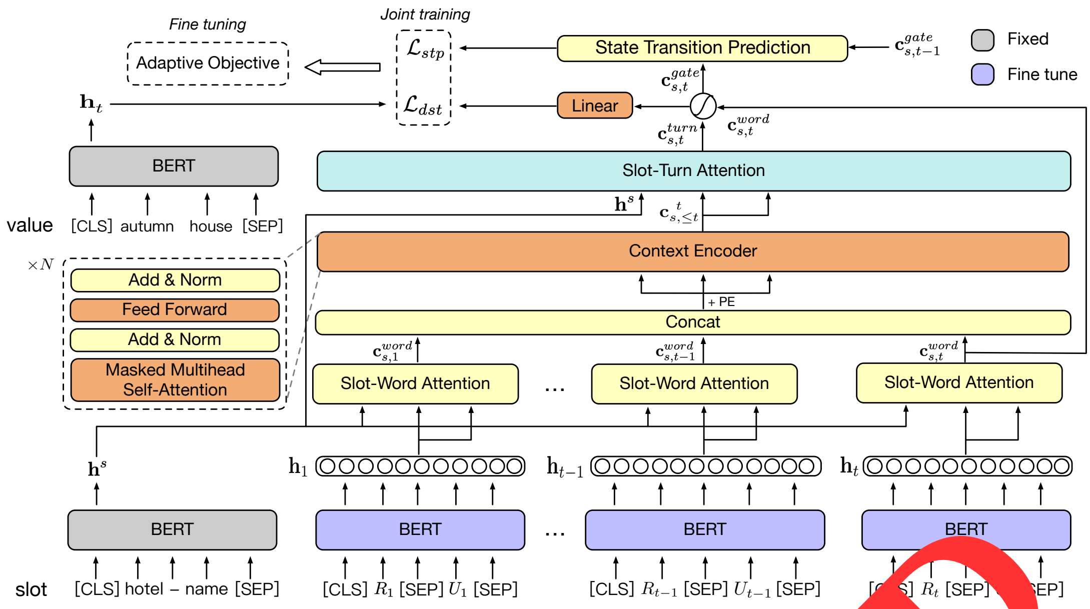

Figure 1: The architecture of our model. At turn t, the slot retrieves relevant information among $\{1, ..., t\}$ turns at both word level and turn level. Specifically, we utilize a context encoder between word level and turn level to capture the relationships between historical relevant information. Finally, we combine the global relevant context $c_{s,t}^{turn}$ and local dialogue information $c_{s,t}^{word}$ as outputs. During training, we first train the DST task and the state transition prediction task jointly, then fine-tune our model with the adaptive objective.

Sentence Encoder. BERT leverages a special token [CLS] to aggregate the whole representation of a sentence and a special token [SEP] to indicate the end of a sentence. For user utterance $U_t = \{w_1^u, ..., w_l^u\}$ and system response $R_t = \{w_1^r, ..., w_{l'}^r\}$ at dialogue turn t, we concatenate them with special tokens and encode them into contextual word representations $\mathbf{h}_t$ as follows:

$$\mathbf{h}_t = \text{BERT}_{finetune}([R_t; U_t]) \quad (1)$$

where $\text{BERT}_{finetune}$ means that it will be fine-tuned during training. Therefore, $\text{BERT}_{finetune}$ will learn a corresponding generalization of sentence representation and adapt to dialogue state tracking task.

For slot s and value v_t, we adopt another pre-trained BERT_{fixed} to encode them into contextual semantics vectors $\mathbf{h}^s$ and $\mathbf{h}_t^v$ respectively. Different from utterances, we use the output vector of the special token [CLS] to obtain the whole sentence representation:

$$\begin{aligned} \mathbf{h}^s &= \text{BERT}_{fixed}(s) \\ \mathbf{h}_t^v &= \text{BERT}_{fixed}(v_t) \end{aligned} \quad (2)$$

where the weights of BERT_{fixed} are fixed during training thus our model can be scalable to any unseen slots and values with sharing the original BERT representation.

Slot-Word Attention. The slot-word attention is a multi-head attention (MultiHead($\mathbf{Q}, \mathbf{K}, \mathbf{V}$)), which takes a query matrix $\mathbf{Q}$, a key matrix $\mathbf{K}$ and a value matrix $\mathbf{V}$ as inputs. Refer to (Vaswani et al., 2017) for more details. For each slot s, the slot-word attention summarizes word-level slot-related information from each turn t into a d-dimensional vector $\mathbf{c}_{s,t}^{word}$, which can be determined as follows:

$$\mathbf{c}_{s,t}^{word} = \text{MultiHead}(\mathbf{h}^s, \mathbf{h}_t, \mathbf{h}_t) \quad (3)$$

Context Encoder. The context encoder is a unidirectional transformer encoder, which is devised to model the contextual relevance of the extracted word-level slot-related information among $\{1, ..., t\}$ turns. The context encoder contains a stack of N identical layers. Each layer has two sub-layers. The first sub-layer is a masked multi-head self-attention (MultiHead), in which $\mathbf{Q} = \mathbf{K} = \mathbf{V}$. The second sub-layer is a position-wise fully connected feed-forward network (FFN), which consists of two linear transformations with a ReLU activation (Vaswani et al., 2017).

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (4)$$

Formally, the output of the context encoder $\mathbf{c}_{s,\leq t}^{ctx}$ can be denoted as follows:

$$\begin{aligned} \mathbf{m}^n &= \text{FFN}(\text{MultiHead}(\mathbf{m}^{n-1}, \mathbf{m}^{n-1}, \mathbf{m}^{n-1})) \\ \mathbf{m}^0 &= [\mathbf{c}_{s,1}^{word} + \text{PE}(1), ..., \mathbf{c}_{s,t}^{word} + \text{PE}(t)] \\ \mathbf{c}_{s,\leq t}^{ctx} &= \mathbf{m}^N \end{aligned} \quad (5)$$

where $\mathbf{m}^n$ is the output of the n-th layer of context encoder and $\mathrm{PE}(\cdot)$ denotes positional encoding function. Note that residual connection and layer normalization are omitted in the formula.

Slot-Turn Attention. To retrieve turn-level relevant information from contextual representation, we devise a slot-turn attention which is the multi-head attention as follows:

$$\mathbf{c}_{s,t}^{turn} = \mathrm{MultiHead}(\mathbf{h}^s, \mathbf{c}_{s,\leq t}^{ctx}, \mathbf{c}_{s,\leq t}^{ctx}) \quad (6)$$

Therefore, the model can access word-level and turn-level relevant information from the historical dialogues.

Global-Local Fusion Gate. To balance the information of global context and local utterances, we propose to dynamically control each proportion of contextual information and current turn information so that the model can not only benefit from relevant context but also keep a balance between global and local representations. Similar to Hochreiter and Schmidhuber (1997), we leverage a fusion gate mechanism, which computes a weight to decide how much global and local information should be combined according to $c_{s,t}^{word}$ and $c_{s,t}^{turn}$. It can be defined as follows:

$$\begin{aligned} g_{s,t} &= \sigma(\mathbf{W}_g \odot [\mathbf{c}_{s,t}^{word}; \mathbf{c}_{s,t}^{turn}]) \\ \mathbf{c}_{s,t}^{gate} &= g_{s,t} \otimes \mathbf{c}_{s,t}^{word} + (1 - g_{s,t}) \otimes \mathbf{c}_{s,t}^{turn} \end{aligned} \quad (7)$$

where $\mathbf{W}_g \in \mathbb{R}^{2d \times d}$ are parameters, σ means sigmoid activation function, $\odot$ and $\otimes$ mean the pointwise and element-wise multiplication respectively.

Finally, we use a linear projection to obtain query results with layer normalization and dropout.

$$\mathbf{o}_{s,t} = \mathrm{LayerNorm}(\mathrm{Linear}(\mathrm{Dropout}(\mathbf{c}_{s,t}^{gate}))) \quad (8)$$

We follow Ren et al. (2018) to adopt L2 norm to compute the distance. Therefore, the probability distribution of value v_t and the training objective can be defined as:

$$p(v_t|U_{\leq t}, R_{\leq t}, s) = \frac{\exp(-\|o_{s,t} - h_t^v\|_2)}{\sum_{v' \in \mathcal{V}_s} \exp(-\|o_{s,t} - h_t^{v'}\|_2)}$$

$$\mathcal{L}_{dst} = \sum_{s \in \mathcal{S}} \sum_{t=1}^{T} -\log(p(\hat{v}_t|U_{\leq t}, R_{\leq t}, s)) \quad (9)$$

where $\mathcal{V}_s$ is the candidate value set of slot s and $\hat{v}_t \in \mathcal{V}_s$ is the ground-truth value of slot s.

2.3 State Transition Prediction

To better capture relevant context, we further introduce an auxiliary binary classification task to jointly train with DST: State Transition Prediction (STP), which is to predict if the value for a slot is updated compared to previous turn. This module reads $\mathbf{c}_{s,t-1}^{gate}$ and $\mathbf{c}_{s,t}^{gate}$ as inputs and the transition probability $p_{s,t}^{stp}$ can be calculated as follows:

$$\begin{aligned} \mathbf{c}_{s,t}^{stp} &= \tanh(\mathbf{W}_c \odot \mathbf{c}_{s,t}^{gate}) \\ p_{s,t}^{stp} &= \sigma(\mathbf{W}_p \odot [\mathbf{c}_{s,t}^{stp}; \mathbf{c}_{s,t-1}^{stp}]) \end{aligned} \quad (10)$$

where $\mathbf{W}_c \in \mathbb{R}^{d \times d}$, $\mathbf{W}_p \in \mathbb{R}^{2d}$ are parameters. Note that when $t = 1$, we simply concatenate $\mathbf{c}_{s,t}^{stp}$ with zero vectors.

For this task, we calculate the binary cross entropy loss between ground-truth transition labels $y_{s,t}^{stp}$ and the transition probability $p_{s,t}^{stp}$, which is defined as follows:

$$\mathcal{L}_{stp} = \sum_{s \in \mathcal{S}} \sum_{t=1}^{T} -y_{s,t}^{stp} \cdot \log(p_{s,t}^{stp}) \quad (11)$$

2.4 Adaptive Objective

Essentially, the slot imbalance problem can be considered as a kind of class imbalance because there is an imbalance among both different slots and different samples. Instead of treating all slots indiscriminately, it is important to balance the learning of different slots. Recently, Lin et al. (2017) propose a soft-sampling method, Focal Loss, to re-weight the losses of different classes.

Inspired by their work, we design a novel adaptive objective for DST which evaluates the difficulty from each slot's accuracy on the validation set and adaptively adjusts the weight of each slot during optimization. We define the accuracy of slot s on validation set as acc_s^{val}. Our adaptive objective is based on the following intuitions:

(1) If $acc_s^{val} \leq acc_{s'}^{val}$; then slot s is more difficult than slot s'. Suppose this slot-level difficulty is defined as α; then

$$\alpha_s = \frac{1 - acc_s^{val}}{\sum_{s' \in \mathcal{S}} 1 - acc_{s'}^{val}} \cdot |\mathcal{S}| \quad (12)$$

(2) Suppose there are two samples $\{(U_t, R_t), (s, v_t)\}$ and $\{(U_{t'}, R_{t'}), (s', v_{t'})\}$. If the former confidence is lower than the latter, then sample $\{(U_t, R_t), (s, v_t)\}$ is more difficult than $\{(U_{t'}, R_{t'}), (s', v_{t'})\}$. Suppose this sample-level difficulty is defined as β; then

$$\beta(s, v_t) = (1 - p(s, v_t))^{\gamma} \quad (13)$$

where $p(s, v_t)$ is the confidence of sample $\{(U_t, R_t), (s, v_t)\}$ and γ is a hyper-parameter.

Thus, the adaptive objective is defined as follows:

$$\mathcal{L}_{adapt}(s, v_t) = -\alpha_s \beta(s, v_t) \log(p(s, v_t)) \quad (14)$$

Focal Loss assigns static learning weights on slots and doesn't change them anymore during the whole training. Compared to Focal Loss, our adaptive objective can fit data better by dynamically evaluate the difficulties in an accuracy-sensitive manner and then adaptively control the learning weights for different slots, which is proved in our experiments. If the difficulty of slot s is greater than the average difficulty of all slots, α_s would increase and enlarge the loss of s. Similarly, the optimization of sample $\{(U_t, R_t), (s, v_t)\}$ with a low confidence $p(s, v_t)$ would be encouraged by a larger loss. When an epoch ends, the adaptive objective re-evaluates the difficulty of each slot and updates α_s. Therefore, it can not only encourage the optimization of those hard slots and samples but also balance the learning of all slots.

2.5 Optimization

In our model, we firstly jointly train the DST and STP tasks to convergence and then fine-tune DST task with the adaptive objective.

During joint training, we optimize the sum of these two loss functions as following:

$$\mathcal{L}_{joint} = \mathcal{L}_{dst} + \mathcal{L}_{stp} \quad (15)$$

At the fine-tuning phase, we adopt the adaptive objective to fine-tune DST task as following:

$$\mathcal{L}_{finetune} = \sum_{s \in \mathcal{S}} \sum_{t=1}^{T} \mathcal{L}_{adapt}(s, \hat{v}_t) \quad (16)$$

3 Experiments Setup

3.1 Datasets & Metrics

	Hotel	Train	Attraction	Restaurant	Taxi
Slots	price, type, parking, stay, day, people, area, stars, internet, name	destination, departure, day, arrive by, leave at, people	area, name, type	food, price, area, name, time, day, people	destination, departure, arrive by, leave by
Train	3381	3103	2717	3813	1654
Valid	416	484	401	438	207
Test	394	494	395	437	195

Table 2: The dataset statistics of MultiWOZ 2.0 & 2.1.

We evaluate our model on MultiWOZ 2.0 (Budzianowski et al., 2018) and MultiWOZ 2.1 (Eric et al., 2019), which are two of the largest public task-oriented dialogue datasets, including about 10,000 dialogues with 7 domains and 35 domain-slot pairs. MultiWOZ 2.1 shares the same dialogues with MultiWOZ 2.0 but it fixed previous annotation errors. The statistics are shown in Table 2. Following (Wu et al., 2019), we use only 5 domains {*restaurant*, *hotel*, *train*, *attraction*, *taxi*} excluding *hospital* and *police* since these two domains never occur in the test set. We preprocess the datasets following (Lee et al., 2019)[2].

We use joint accuracy and slot accuracy as our evaluation metrics. Joint accuracy is the accuracy of the dialogue state of each turn and a dialogue state is evaluated correctly only if all the values of slots are correctly predicted. Slot accuracy only considers individual slot-level accuracy.

3.2 Baseline Models

We compare our results with the following competitive baselines:

DSTreader proposes to model DST as a machine reading comprehension task and extract spans from dialogue history (Gao et al., 2019b).

GLAD-RCFS uses a heuristic rule to extract relevant turns and lets slot-value pairs to query relevant context from them (Sharma et al., 2019).

HyST employs a hierarchical encoder and takes a hybrid way combining both predefined-ontology and open-vocabulary settings (Goel et al., 2019).

TRADE encodes the whole dialogue context and decodes the value for every slot using a copy-augmented decoder (Wu et al., 2019).

DST-QA proposes to model DST as a question answering problem and uses a dynamically-evolving knowledge graph to learn relationships between slot pairs (Zhou and Small, 2019).

SOM-DST considers the dialogue state as an explicit fixed-size memory and proposes a selectively overwriting mechanism (Kim et al., 2019).

SUMBT exploits BERT as the encoder of the utterances, slots and values. It scores every candidate slot-value pair in a non-parametric manner using a distance measurement (Lee et al., 2019).

DST-picklist performs matchings between candidate values and slot-context encoding considering all slots as picklist-based slots (Zhang et al., 2019).

GLAD-RCFS, HySST, SUMBT, DST-picklist are predefined-ontology models as well as our model and DSTreader, TRADE, DST-QA, SOM-DST are open-vocabulary models.

[2]https://github.com/SKTBrain/SUMBT

Model	Ontology	MultiWOZ 2.0		MultiWOZ 2.1	
		Joint (%)	Slot (%)	Joint (%)	Slot (%)
DSTreader (Gao et al., 2019b)	×	39.41	-	36.40*	-
GLAD-RCFS (Sharma et al., 2019)	✓	46.31	-	-	-
HyST (Goel et al., 2019)	✓	42.33	-	38.10*	-
TRADE (Wu et al., 2019)	×	48.60	96.92	45.60*	-
DST-QA (Zhou and Small, 2019)	×	51.44	97.24	51.17	97.21
SOM-DST (Kim et al., 2019)	×	51.38	-	52.57	-
SUMBT (Lee et al., 2019)	✓	48.81†	97.33†	52.75‡	97.56‡
DST-picklist (Zhang et al., 2019)	✓	-	-	53.30	-
Our Model	✓	**52.68**	**97.69**	**58.55**	**98.14**

Table 3: Joint accuracy & slot accuracy on the test sets of MultiWOZ 2.0 and 2.1. The ontology column indicates if a model is based on predefined ontology or not. † means the updated results on SUMBT's GitHub[2] and ‡ means our reproduction results using source code of SUMBT [2]. * means we borrow results from (Eric et al., 2019).

3.3 Settings

We employ the pre-trained BERT model that has 12 layers of 784 hidden units and 12 self-attention heads [3]. For the multi-head attention, we set heads count and hidden size to 4 and 784, respectively. For the context encoder, we set the transformer layers to 6. We set the max sequence length of all inputs to 64 and the batch size to 32. In all training, we use Adam optimizer (Kingma and Ba, 2015) and set the warmup proportion to 0.1. Specifically, in the joint training phase, we set the peak learning rate to 1e-4. At the fine-tuning phase, we set γ to 2, peak learning rate to 1e-5. The training stopped early when the validation loss was not improved for 15 consecutive epochs. For all experiments, we report the mean joint accuracy over multiple different random seeds to reduce statistical errors.

4 Experiment Results

4.1 Main Results

Table 3 shows the joint accuracy of our model and other baselines on the test sets of MultiWOZ 2.0 and 2.1. Our model beats all baselines whether they are based on predefined ontology or open vocabulary, and achieves 52.68% and 58.55% joint accuracy with considerable improvements (1.24% and 5.98%) over previous best results on MultiWOZ 2.0 and 2.1, respectively. Also, our model achieves 97.69% and 98.14% slot accuracy with 0.36% and 0.58% improvements over the previous best results on MultiWOZ 2.0 and 2.1, respectively. Similar to (Kim et al., 2019), we find that our model achieves much higher improvements on MultiWOZ 2.1 than on MultiWOZ 2.0. This is probably because MultiWOZ 2.1 fixes lots of notation errors in MultiWOZ 2.0 and our model can benefit more from more accurate relevant context.

[3]It is published as *bert-base-uncased* model in https://github.com/huggingface/pytorch-transformers

Model	MultiWOZ 2.1
Our Model	**58.55**
- state transition prediction	57.86 (-0.69)
- adaptive objective fine-tuning	57.45 (-1.10)
- above two (only CHAN)†	57.00 (-1.55)
Our Model-FL (α=1,γ=2)‡	58.10 (-0.45)

Table 4: The ablation study of the state transition prediction and the adaptive objective on the MultiWOZ 2.1 test set with joint accuracy (%). † means removing above two modules and remaining CHAN only. ‡ means fine-tuning with focal loss instead.

4.2 Ablation Study

As shown in Table 4, we estimate the effectiveness of the proposed state transition prediction and adaptive objective on the MultiWOZ 2.1 test set. The results show that both state transition prediction task and adaptive objective can boost the performance. Removing the state transition prediction task reduces joint accuracy by 0.69%, and the joint accuracy decreases by 1.10% without the adaptive objective fine-tuning. Moreover, when we remove the state transition prediction task and don't fine-tune our model with adaptive objective (only CHAN remains), the joint accuracy decreases by 1.55%. Also, to explore the importance of adjusting the α_s adaptively, we replace the adaptive objective with original focal loss ($\alpha = 1, \gamma = 2$), which leads to 0.45% drop.

To prove the effectiveness of each module of the proposed CHAN, we conduct ablation experiments

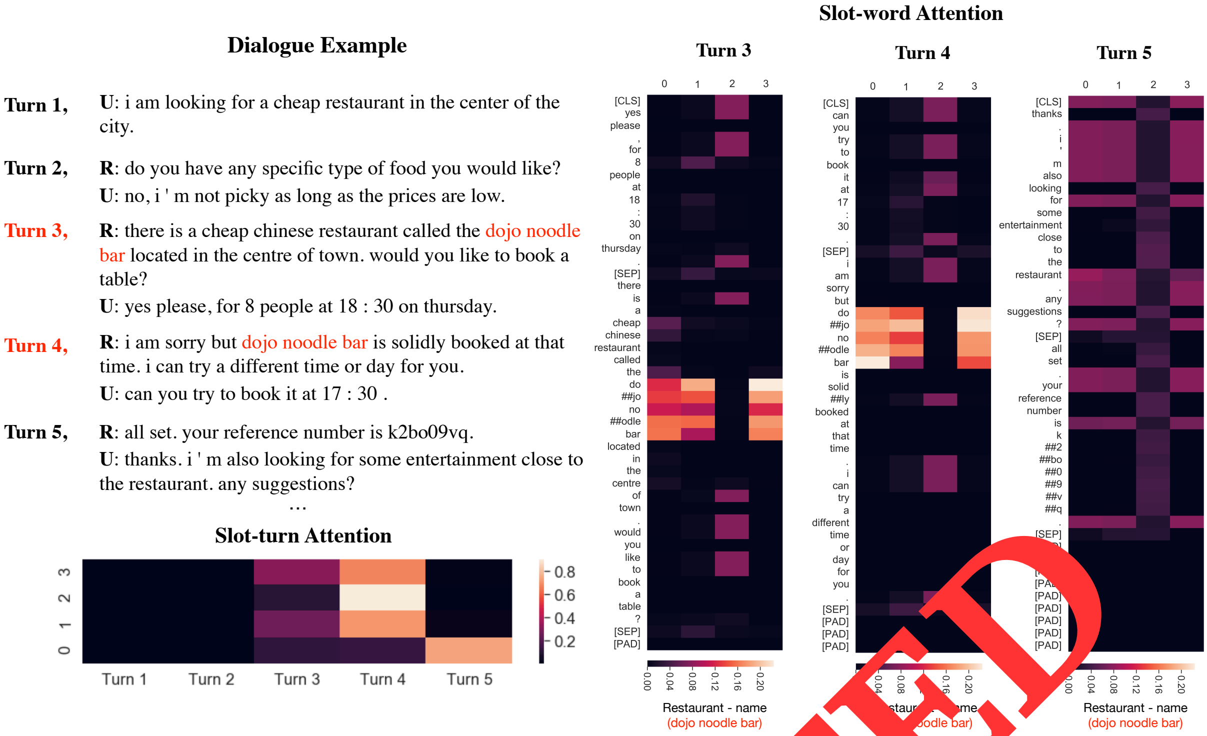

Figure 2: The turn-level and word-level attention visualization of our model on an example from MultiWOZ 2.1 test set, which is predicting the value of slot "*restaurant-name*" at the 5th turn. The columns "0,1,2,3" are the index of each head of multi-head attention. Although there is no slot-related information at 5th turn, our model still makes the correct prediction by attending to historiacal relevant words "*dojo noodle bar*" and relevant turns {3,4}, which is highlighted in red. Best viewed in color.

on the MultiWOZ 2.1 test set as shown in Table 5. We observe that a slight joint accuracy drop of 0.24% after removing the global-local fusion gate, which proves the effectiveness of fusing global context and local utterances. Moreover, removing the slot-turn attention and context encoder leads to a decrease by 0.15% and 1.72% respectively, which demonstrates that the turn-level relevant information and the contextual representations of word-level relevant information are effective to improve the performance. Moreover, after we remove the aforementioned three modules and sum the word-level relevant information of $\{1, \cdots, t\}$ turns as output, the joint accuracy reduces by 6.72%, which is much higher than the sum of above three reductions. It demonstrates that effectively modeling interactions with word-level relevant information of dialogue history is crucial for DST.

Model	MultiWOZ 2.1
CHAN	**57.00**
- global-local fusion gate	56.76 (-0.24)
- slot-turn attention	56.85 (-0.15)
- context encoder	55.28 (-1.72)
- above three†	50.28 (-6.72)

Table 5: The ablation study of the CHAN on the MultiWOZ 2.1 test set with joint accuracy (%). † means removing above three modules and summing the word-level relevant information of $\{1, \cdots, t\}$ turns as output.

4.3 Attention Visualization

Figure 2 shows the visualization of turn-level and word-level attention of the "*restaurant-name*" slot on a prediction example of our model at turn 5. The turn-level attention visualization indicates that our model attends to the turns $\{3, 4\}$ that are semantically related to the given slots "*restaurant-name*" while almost pays no attention to turns {1,2}. And from the word-level attention visualization, we can easily find that the "*restaurant-name*" slot attends to the "*dojo noodle bar*" with the highest weight in both turn 3 and turn 4. Although there is no slot-related information at turn 5, our model still makes the correct decision by exploiting relevant context from the historical dialogue.

4.4 Effects of Adaptive Obj. on Acc. per Slot

As Figure 3 shows, we draw the accuracy changes of each slot on MultiWOZ 2.1 test set after fine-tuning our model with adaptive objective. We sort all slots in ascending order according to their fre-

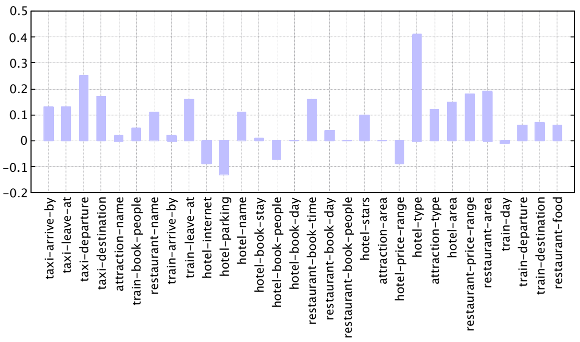

Figure 3: The accuracy changes (%) of each slot on the MultiWOZ 2.1 test set after fine-tuning with adaptive objective. We sort all slots in ascending order according to their frequency (Please refer to Appendix for detailed accuracy results).

quency (The detailed accuracy results are in the Appendix). Thus, slots on the left side are relatively more difficult than slots on the right side. After fine-tuning with the adaptive objective, most slots on the left side achieve significant improvements, which proves the adaptive objective can encourage the learning of the hard slots. Although adaptive objective tends to decrease the weight of slots on the right side, they also benefit from the fine-tuning. We think that this is because encouraging the optimizing of hard slots enhances our model by tracking more complicated dialogue states. It proves that our adaptive objective can not only improve the performance of relatively hard slots but also boost the performance of relatively easy slots.

4.5 Qualitative Analysis

To explore the advantages of our model compared to baseline models, we conduct a human evaluation on a subset of the MultiWOZ 2.1 test set where our model makes correct predictions while SUMBT (a previous strong baseline) fails. We predefine three types of improvements: historical information inference improvement which means inferring historical information is necessary for correct decisions, current information inference improvement which means inferring current information is enough for correct decisions, and other improvements. As shown in Table 6, 64.49% improvements come from historical information inference, which demonstrates that our model can better exploit relevant context from the dialogue history.

Improvement Type	Percentage
Historical Information Inference Improvement	64.49%
Current Information Inference Improvement	34.86%
Others	0.65%

Table 6: Qualitative analysis on the improvements of our model compared to a previous strong baseline SUMBT. It is evaluated by human on a subset of MultiWOZ 2.1 test set where our model makes correct predictions while SUMBT fails.

5 Related Work

Traditional statistical dialogue state tracking models combine semantics extracted by spoken language understanding modules to predict the current dialogue state (Williams and Young, 2007; Thomson and Young, 2010; Wang and Lemon, 2013; Williams, 2014) or to jointly learn speech understanding (Henderson et al., 2014b; Zilka and Jurcicek, 2015; Wen et al., 2017). One drawback is that they rely on hand-crafted features and complex domain-specific lexicons besides the ontology, and they are hard to extend and scale to new domains. Recent neural network models are proposed for further improvements (Mrkšić et al., 2015; Hori et al., 2016; Mrkšić et al., 2017; Lei et al., 2018; Xu and Hu, 2018; Zhong et al., 2018; Nouri and Hosseini-Asl, 2018; Wu et al., 2019; Ren et al., 2019; Balaraman and Magnini, 2019). Ren et al. (2018) and Lee et al. (2019) use an RNN to encode the slot-related information of each turn, where slots can not attend to relevant information of past turns directly. Sharma et al. (2019) employ a heuristic rule to extract partial dialogue history and then integrate the historical information into prediction in a coarse manner. Goel et al. (2019) encode the dialogue history into a hidden state and then simply combine it with the slot to make decisions. These models are deficient in fully exploiting the relevant context in dialogue history.

Gao et al. (2019b) introduce a slot carryover model to decide whether the values from the previous turn should be used or not and Kim et al. (2019) introduce a state operation predictor to decide the operation with the previous state. Different from them, we consider the state transition prediction as an additional enhancement while they integrate it into their DST pipelines. Besides, Zhong et al. (2018) only employ local modules to model the slot-specific representations, which neglects the slot imbalance problem.

The general backbone of our model is a hierarchical attention network that can effectively aggregate query-related information at multiple levels (Yang et al., 2016; Ying et al., 2018; Wang et al., 2018; Xing et al., 2018; Aujogue and Aussem, 2019; Naik et al., 2018; Liu and Chen, 2019).

6 Conclusion

We introduce an effective model that consists of a contextual hierarchical attention network to fully exploit relevant context from dialogue history and an adaptive objective to alleviate the slot imbalance problem in dialogue state tracking. Experimental results show that our model achieves state-of-the-art performance of 52.68% and 58.55% joint accuracy with considerable improvements (+1.24% and +5.98%) over previous best results on MultiWOZ 2.0 and MultiWOZ2.1 datasets, respectively.

Although our model is based on predefined ontology, it is universal and scalable to unseen domains, slots and values. The main contributions of our model, CHAN and adaptive objective, can also be applied to open-vocabulary models. We will explore it in the future.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. This work was supported by National Key R&D Program of China (NO. 2017YFE0192900).

References

Jean-Baptiste Aujogue and Alex Aussem. 2019. Hierarchical recurrent attention networks for context-aware education chatbots. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Vevake Balaraman and Bernardo Magnini. 2019. Scalable neural dialogue state tracking. *arXiv preprint arXiv:1910.09942*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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.

Guan-Lin Chao and Ian Lane. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. *Proc. Interspeech 2019*, pages 1468–1472.

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.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Jianfeng Gao, Michel Galley, Lihong Li, et al. 2019a. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, Dilek Hakkani-Tur, and Amazon Alexa AI. 2019b. Dialog state tracking: A neural reading comprehension approach. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 264.

Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. 2019. Hyst: A hybrid approach for flexible and accurate dialogue state tracking. *Proc. Interspeech 2019*.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. 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)*, pages 292–299.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Takaaki Hori, Hai Wang, Chiori Hori, Shinji Watanabe, Bret Harsham, Jonathan Le Roux, John R Hershey, Yusuke Koji, Yi Jing, Zhaocheng Zhu, et al. 2016. Dialog state tracking with attention-based sequence-to-sequence learning. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 552–558. IEEE.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2019. Efficient dialogue state tracking by selectively overwriting memory. *arXiv preprint arXiv:1911.03906*.

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*.

Hung Le, Doyen Sahoo, Chenghao Liu, Nancy F. Chen, and Steven C.H. Hoi. 2020a. End-to-end multi-domain task-oriented dialogue systems with multi-level neural belief tracker. In *OpenReview*.

Hung Le, Richard Socher, and Steven C.H. Hoi. 2020b. Non-autoregressive dialog state tracking. In *International Conference on Learning Representations*.

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.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Zhengyuan Liu and Nancy Chen. 2019. Reading turn by turn: Hierarchical attention architecture for spoken dialogue comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5460–5466.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Multi-domain dialog state tracking using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 794–799.

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.

Chetan Naik, Arpit Gupta, Hancheng Ge, Mathias Lambert, and Ruhi Sarikaya. 2018. Contextual slot carryover for disparate schemas. *Proc. Interspeech 2018*, pages 596–600.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking model. In *Proceedings of NeurIPS 2018, 2nd Conversational AI workshop*.

Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1876–1885.

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.

Sanuj Sharma, Prafulla Kumar Choubey, and Ruihong Huang. 2019. Improving dialogue state tracking by discerning the relevant context. 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 576–581.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wei Wang, Ming Yan, and Chen Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1705–1714.

Zhuoran Wang and Oliver Lemon. 2013. 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*, pages 423–432.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. 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*, pages 438–449.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413.

Jason D Williams. 2014. Web-style ranking and slu combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 282–291.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 808–819.

Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. 2018. Hierarchical recurrent attention network for response generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential recommender system based on hierarchical attention networks. In *the 27th International Joint Conference on Artificial Intelligence*.

Steve Young. 2002. Talking to machines (statistically speaking). In *Seventh International Conference on Spoken Language Processing*.

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

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.

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.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*.

Lukas Zilka and Filip Jurcicek. 2015. Incremental lstm-based dialog state tracker. In *2015 Ieee Workshop on Automatic Speech Recognition and Understanding (Asru)*, pages 757–762. IEEE.

A Slot Imbalance

Figure 4 shows the relationships between frequency and accuracy of slots (left) and slot-value pairs (right). Because the frequency will be the same for all slots if we consider "*none*" as well, we calculate accuracy with "*none*" value excluded for slots. Overall, the more the frequency, the higher the accuracy. It demonstrates that the slot imbalance problem results in different learning difficulties for different slots. Moreover, the slot imbalance problem makes some slots hard to learn and hence hurts the accuracy, which limits the overall performance.

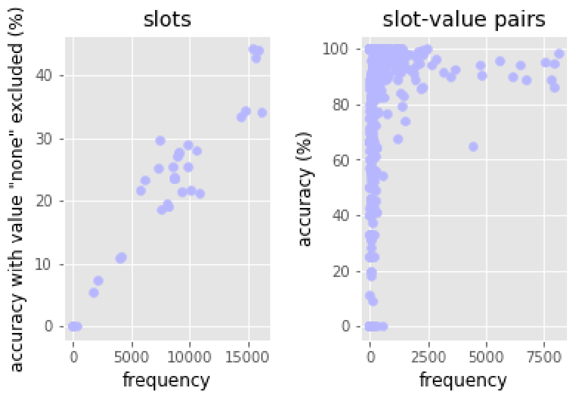

Figure 4: The relationships between frequency and accuracy of slots (left) and slot-value pairs (right). Because the frequency will be the same for all slots if we consider "*none*" as well, we calculate accuracy with "*none*" value excluded for slots.

B Acc. per Slot on MultiWOZ 2.1 Testset

Domain-Slot	Frequency	Our Model without adaptive objective	Our Model	Δ
taxi-arrive by	1794	99.13	99.25	0.13
taxi-leave at	2165	99.14	99.27	0.13
taxi-departure	4037	98.12	98.37	0.25
taxi-destination	4108	98.1	98.26	0.17
attraction-name	5843	94.16	94.18	0.02
train-book people	6178	97.72	97.76	0.05
restaurant-name	7293	93.67	93.78	0.11
train-arrive by	7488	97.97	97.99	0.02
train-leave at	7563	96.05	96.22	0.16
hotel-internet	8012	97.26	97.16	-0.09
hotel-parking	8179	97.28	97.14	-0.13
hotel-name	8621	95.41	95.52	0.11
hotel-book stay	8715	99.44	99.46	0.01
hotel-book people	8734	99.35	99.28	-0.07
hotel-book day	8745	99.28	99.28	0
restaurant-book time	8958	99.15	99.3	0.16
restaurant-book day	9021	99.31	99.35	0.04
restaurant-book people	9026	99.35	99.35	0
hotel-stars	9330	98.31	98.41	0.1
attraction-area	9766	[illegible]	98.03	0
hotel-price range	9793	98.69	98.6	-0.09
hotel-type	10110	[illegible]	[illegible]	0.41
attraction-type	10525	[illegible]	[illegible]	0.12
hotel-area	10885	[illegible]	97.67	0.15
restaurant-price range	[illegible]	97.66	97.84	0.18
restaurant-area	[illegible]	97.68	97.86	0.19
train-day	[illegible]	[illegible]	99.42	-0.01
train-departure	15672	[illegible]	98.48	0.06
train-destination	15951	[illegible]	98.7	0.07
restaurant-food	16095	97.54	97.61	0.06

Table 7: The detailed results of accuracy (%) per slot before and after fine-tuning our model with adaptive objective on MultiWOZ 2.1 test set. We sort them in ascending order according to their frequency. Δ means the changes of accuracy after fine-tuning.