MinTL: Minimalist Transfer Learning for Task-Oriented Dialogue Systems

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
zlinao@connect.ust.hk

Abstract

In this paper, we propose Minimalist Transfer Learning (*MinTL*) to simplify the system design process of task-oriented dialogue systems and alleviate the over-dependency on annotated data. *MinTL* is a simple yet effective transfer learning framework, which allows us to plug-and-play pre-trained seq2seq models, and jointly learn dialogue state tracking and dialogue response generation. Unlike previous approaches, which use a copy mechanism to "carryover" the old dialogue states to the new one, we introduce *Levenshtein belief spans* (Lev), that allows efficient dialogue state tracking with a minimal generation length. We instantiate our learning framework with two pre-trained backbones: T5 (Raffel et al., 2019) and BART (Lewis et al., 2019), and evaluate them on MultiWOZ. Extensive experiments demonstrate that: 1) our systems establish new state-of-the-art results on end-to-end response generation, 2) *MinTL*-based systems are more robust than baseline methods in the low resource setting, and they achieve competitive results with only 20% training data, and 3) Lev greatly improves the inference efficiency[1].

1 Introduction

Building robust task-oriented dialogue systems is challenging due to complex system design and limited availability of human-annotated data (Wen et al., 2017; Wu et al., 2019b). A dialogue agent is expected to learn dialogue reasoning, decision making, and language generation, which require a large amount of training data. However, collecting and annotating data for training a dialogue system is time-intensive and not transferable among domains (Young et al., 2013). One possible workaround is to leverage the pre-trained language model to reduce human supervision (Budzianowski and Vulić, 2019).

Recent progress in pre-training language models has been shown to be promising in alleviating the data scarcity problem (Budzianowski and Vulić, 2019; Wu et al., 2020). Such models are typically pre-trained on large-scale plain text with self-supervised objectives, e.g., language modeling (Radford et al., 2019) and language denoising (Devlin et al., 2019). Fine tuning pre-trained language models improves a wide range of natural language processing applications (Lewis et al., 2019; Raffel et al., 2019), notably machine translation (Conneau and Lample, 2019), and personalized dialogue response generation (Wolf et al., 2019b). However, adapting pre-trained language models to task-oriented dialogue systems is not trivial. Current state-of-the-art (SOTA) approaches in task-oriented dialogue rely on several tasks-specific modules, such as State Operation Predictor (Kim et al., 2019) for dialogue state tracking, and CopyNet (Gu et al., 2016) for end-to-end dialogue task completion (Lei et al., 2018; Zhang et al., 2019b). Such modules are usually absent in the pre-training stage. Therefore, tasks-specific architecture modifications are required in order to adapt pre-trained language models to different dialogue tasks.

In this work, we aim to simplify the process of transferring the prior knowledge of pre-trained language models for improving task-oriented dialogue systems. We propose Minimalist Transfer Learning (*MinTL*), a simple yet effective transfer learning framework that allows to plug-and-play pre-trained sequence-to-sequence (Seq2Seq) models and jointly learn dialogue state tracking (DST) and dialogue response generation. Unlike previous approaches (Lei et al., 2018; Zhang et al., 2019b), which use a copy mechanism to "carryover" the previous dialogue states and generate new dialogue states, we introduce *Levenshtein belief spans* (Lev)

[1]Code available in https://github.com/zlinao/MinTL

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 3391–3405,
November 16–20, 2020.

which models the difference between old states and new states. In practice, *MinTL* first decodes the Lev for updating the previous dialogue state; then, the updated state is used to search the external knowledge base; and finally, a response decoder decodes response by conditioning on the dialogue context and knowledge base match result.

MinTL is easy to set up by using different pre-trained seq2seq backbones. We conduct extensive experiments on both DST and end-to-end dialogue response generation tasks with two pre-trained seq2seq models, such as T5 (Raffel et al., 2019) and BART (Lewis et al., 2019). The experimental result on a large-scale task-oriented dialogue benchmark MultiWOZ (Budzianowski et al., 2018; Eric et al., 2019) suggests that our proposed method significantly improves SOTA performance in both the full data and simulated low resource setting. Our contributions are summarized as follows:

- We propose the *MinTL* framework that efficiently leverages pre-trained language models for task-oriented dialogue without any ad hoc module.
- We propose the novel Lev for efficiently tracking the dialogue state with the minimal length of generation, which greatly reduces the inference latency.
- We instantiate our framework with two different pre-trained backbones, and both of them improve the SOTA results by a large margin.
- We demonstrate the robustness of our approach in the low-resource setting. By only using 20% training data, *MinTL*-based systems achieve competitive results compared to the SOTA.

2 Related Work

Pre-trained Language Models. Language model (LM) pre-training (Radford et al., 2019; Devlin et al., 2019; Yang et al., 2019), has been shown to be beneficial in NLP downstream tasks. Generative pre-trained unidirectional LMs (e.g., GPT2) are effective in language generation tasks (Radford et al., 2019; Hosseini-Asl et al., 2020; Peng et al., 2020; Lin et al., 2020). Several works have applied a generative pre-training approach in open domain chitchat tasks (Wolf et al., 2019b; Zhang et al., 2019c), and achieved promising results. On the other hand, bidirectional pre-trained LMs (Devlin et al., 2019; Liu et al., 2019) significantly improve the performance of natural language understanding tasks. These models are usually evaluated on classification tasks such as the GLUE benchmark (Wang et al., 2018), extractive question answering tasks (Rajpurkar et al., 2016), and dialogue context understanding (Wu et al., 2020). However, their bidirectionality nature makes them difficult to be applied to natural language generation tasks (Dong et al., 2019). Recent works (Dong et al., 2019; Raffel et al., 2019; Lewis et al., 2019) unified unidirectional LM and bidirectional LM pre-training approaches, and proposed a Seq2Seq LM, which are pre-trained with language denoising objectives. A systematic study conducted by Raffel et al. (2019) suggests that the combination of an encoder-decoder architecture and language denoising pre-training objectives yields the best result in both language understanding and generation tasks. Notably, the two latest pre-trained chatbots, Meena (Adiwardana et al., 2020) and BST (Roller et al., 2020), are also built on an encoder-decoder architecture. In this work, we transfer the prior knowledge of Seq2Seq LMs to task-oriented dialogues, and successfully improve the SOTA (Zhang et al., 2019b) result with less human annotation.

Task-Oriented Dialogue. Task-oriented dialogue systems are designed to accomplish a goal described by a user in natural language. Such systems are usually built with a pipeline approach. The pipeline often requires natural language understanding (NLU) for belief state tracking, dialogue management (DM) for deciding which actions to take, and natural language generation (NLG) for generating responses (Williams and Young, 2007). To simplify the system design and reduce human supervision, several end-to-end trainable systems have been proposed (Bordes et al., 2016; Wen et al., 2017; Lei et al., 2018; Neelakantan et al., 2019; Eric and Manning, 2017; Eric et al., 2017; Madotto et al., 2018). These methods have been shown to achieve promising results in single-domain tasks. However, the recently proposed multi-domain task-oriented dialogue datasets (Budzianowski et al., 2018; Eric et al., 2019) bring new challenges for multi-domain dialogue state tracking and response generation. Several follow up works (Wu et al., 2019a; Chen et al., 2019; Budzianowski and Vulić,

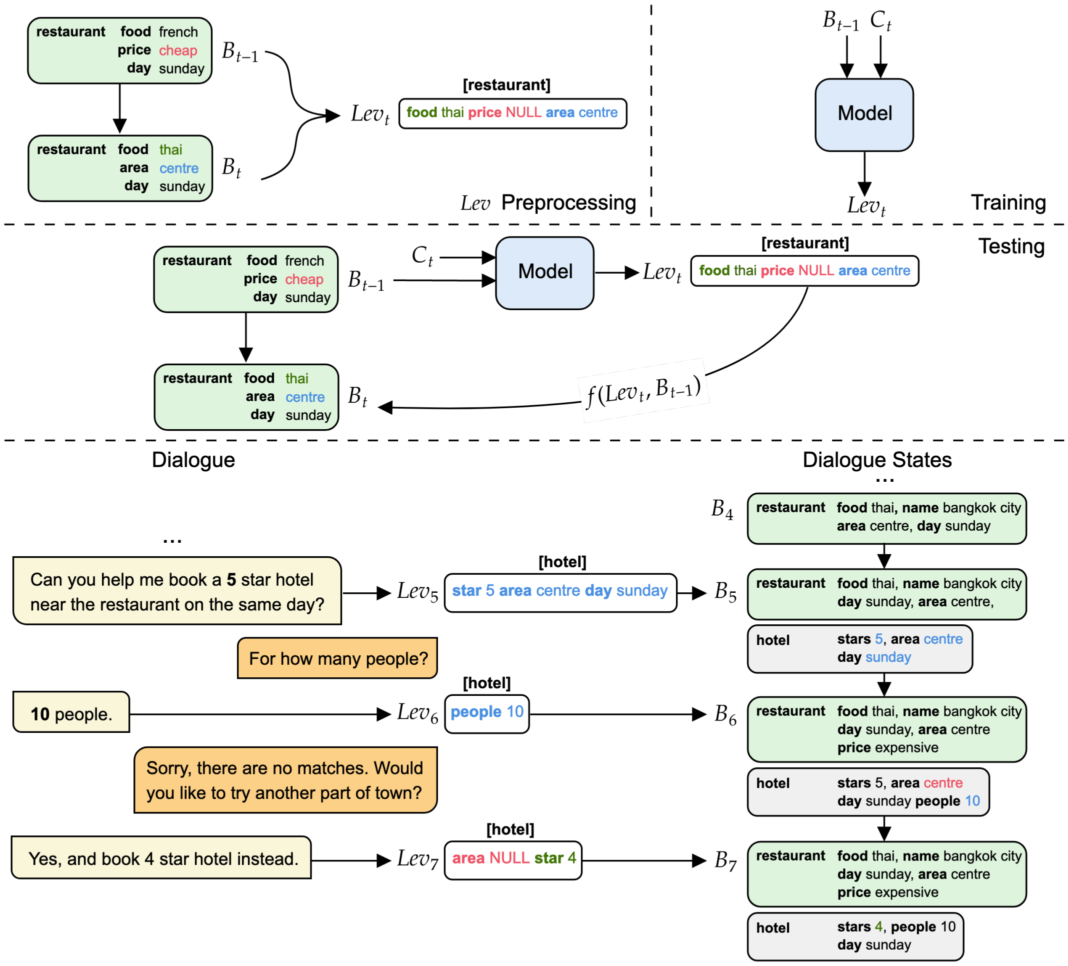

Figure 1: Dialogue state tracking with Lev. The model first generates Lev, then updates the dialogue state with new generated slot-values. The updating operations are insertion (blue), deletion (red), and substitution (green).

2019; Mehri et al., 2019; Madotto et al., 2020b) improved on the initial baselines with various methodologies. Zhang et al. (2019b) proposed the domain aware multi-decoder network and augmented the system act labels by leveraging the user act annotation, achieving the SOTA results in MultiWoz. However, the aforementioned works rely on task-specific design and extensive human annotations. To reduce the human effort and simplify the system design, we propose a simple transfer learning framework that can be easily set up with pre-trained Seq2Seq models and obtain decent performance with a small fraction of the training data.

3 Methodology

In this section, we first provide the notations that are used throughout the paper, then we introduce the Lev for efficient DST, and finally, describe the *MinTL* framework and two backbone models.

Notations. Let us define a dialogue $\mathcal{C} = \{U_1, R_1, \dots, U_T, R_T\}$ as an alternating set of utterances from two speakers, where U and R represent the user utterance and the system response, respectively. At turn t, we denote a dialogue context as $C_t = \{U_{t-w}, R_{t-w}, \dots, R_{t-1}, U_t\}$ and system response as R_t, where w is the context window size. $\mathcal{B} = \{B_1, \dots, B_T\}$ is the dialogue states for each turn. We define B_t, the dialogue state at turn t, as a dictionary that maps (domain: d_i, slot: s_j) a pair into values v, where $\mathcal{D} = \{d_1, \dots, d_N\}$ are the domains, and $\mathcal{S} = \{s_1, \dots, s_M\}$ are slots to track. Thoughtout the paper, we denote the value of a pair (d_i, s_j) in B_t as $B_t(d_i, s_j) = v$, and $B_t(d_i, s_j) = \varepsilon$ when key (d_i, s_j) is not in B_t, where ε denotes an empty string, and $|\varepsilon| = 0$.

3.1 Levenshtein Belief Spans

The goal of DST is to track the slot values for each domain mentioned in dialogue. Existing works either perform classifications for each slot over a

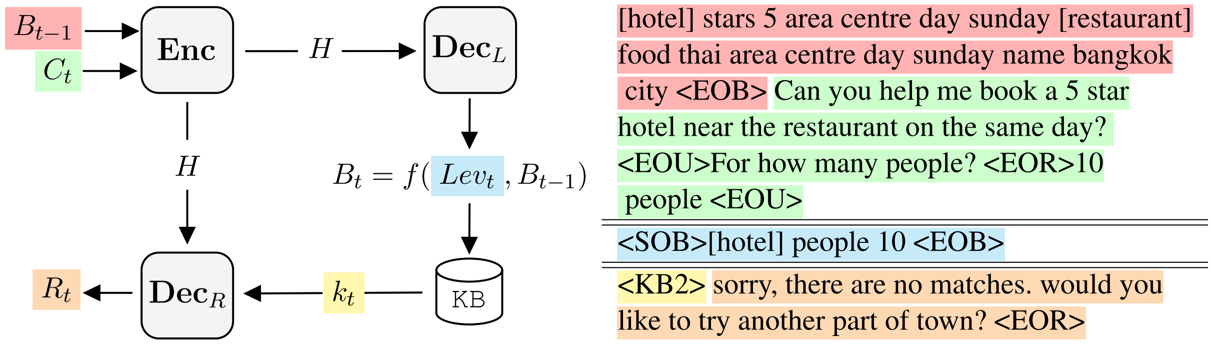

Figure 2: Overview of the *MinTL* framework. The left figure shows the information flow among all modules. The explicit inputs and outputs of each module are described on the right. *MinTL* first encodes previous dialogue state B_t and dialogue context C_t, and decodes Lev_t. Then Lev_t is used to update B_{t-1} to B_t via function f. The updated B_t is used to query the KB and booking API and return KB state k_t. Finally, the R_t is generated by conditioning on B_{t-1}, C_t and k_t.

candidate-value list (Zhang et al., 2019a) or directly generate slot values with a generative model (Lei et al., 2018; Wu et al., 2019a; Kim et al., 2019; Le et al., 2020). Notably, Lei et al. (2018) introduce the concept of *Belief span* that reformats the dialogue states into a text span for allowing models to generate slot values dynamically. Compared to classification based DST, generative DST models can predict the slot values without full access to predefined ontology. However, the aforementioned generative methods either generate the belief span from scratch (Lei et al., 2018) or classify the state operations over all the combinations of domain slot pairs for decoding necessary slot values (Kim et al., 2019; Le et al., 2020), which is not scalable when interfacing to a large number of services and APIs spanning multiple domains (Rastogi et al., 2019).

The idea of Lev is to generate minimal belief spans at each turn for editing the previous dialogue states. As illustrated in Figure 1, Lev is constructed at training time as the DST training target. Given B_{t-1}, B_t, and a pair of (d_i, s_j), we define the three slot level edit operation conditions, i.e., insertion (`INS`), deletion (`DEL`) and substitution (`SUB`), as:

$$\texttt{INS} \rightarrow B_t(d_i, s_j) \neq \varepsilon \wedge B_{t-1}(d_i, s_j) = \varepsilon \quad (1)$$

$$\texttt{DEL} \rightarrow B_t(d_i, s_j) = \varepsilon \wedge B_{t-1}(d_i, s_j) \neq \varepsilon \quad (2)$$

$$\texttt{SUB} \rightarrow B_t(d_i, s_j) \neq B_{t-1}(d_i, s_j). \quad (3)$$

In domain d_i, to update the $B_{t-1}(d_i, s_j)$ to $B_t(d_i, s_j)$, the minimal slot-value pair needed to be generated is $E(d_i, s_j)$, defined as

$$E(d_i, s_j) = \begin{cases} s_j \oplus B_t(d_i, s_j) & \text{if } \texttt{INS} \\ s_j \oplus \texttt{NULL} & \text{if } \texttt{DEL} \\ s_j \oplus B_t(d_i, s_j) & \text{if } \texttt{SUB} \\ \varepsilon & \text{otherwise,} \end{cases} \quad (4)$$

where $\oplus$ denotes string concatenation. `NULL` is the symbol denoting to delete the slot (d_i, s_j) from B_{t-1}. Then, we aggregate all the $E(d_i, s_j)$ for domain d_i as follows:

$$L(d_i) = E(d_i, s_1) \oplus \cdots \oplus E(d_i, s_M). \quad (5)$$

When the dialogue state of domain d_i needs to be updated, i.e., $L(d_i) \neq \varepsilon$, we append the domain information $[d_i]$ at the beginning of $L(d_i)$ to construct Lev of domain d_i:

$$\delta(L, d_i) = \begin{cases} [d_i] \oplus L(d_i) & \text{if } L(d_i) \neq \varepsilon \\ \varepsilon & \text{otherwise.} \end{cases} \quad (6)$$

Finally, we formally define Lev as the following:

$$Lev = \delta(L, d_1) \oplus \cdots \oplus \delta(L, d_N). \quad (7)$$

At inference time, the model first generates Lev_t at turn t, then edits the B_{t-1} by using a deterministic function f, defined as:

$$B_t = f(Lev_t, B_{t-1}). \quad (8)$$

This function simply update the B_{t-1} when new slot-value pairs appear in Lev_t, and it delete the corresponding slot-value when the `NULL` symbol is generated.

Figure 1 shows an example of editing the dialogue state editing process using Lev. In the 6-th turn, the generated Lev_6 inserts the value 10 into the slot *people*. In the 7-th turn, the `NULL` in Lev_7 triggers the `DEL` operation, and thus the slot $(hotel, area)$ is deleted in B_6, which is equivalent to $B_7(hotel, area) = \varepsilon$.

3.2 *MinTL* Framework

Figure 2 describes the flow of the *MinTL* framework with a general encoder-decoder architecture. The input of our framework is a dialogue context C_t and a previous dialogue state B_{t-1}. All sub-sequences are concatenated with special segment tokens, i.e., B_{t-1}<EOB>... R_{t-1}<EOR>U_t<EOU>, as input to the encoder.

$$H = Encoder(\mathcal{C}_t, B_{t-1}), \tag{9}$$

where the $H \in \mathbb{R}^{I \times d_{model}}$ is the hidden states of the encoder, and I is the input sequence length. Then, the Lev decoder attends to the encoder hidden states H and decodes Lev_t sequentially:

$$Lev_t = Decoder_L(H). \tag{10}$$

The learning objective of this generation process is minimizing the negative log-likelihood of Lev_t given $\mathcal{C}_t$ and B_{t-1}, that is

$$\mathcal{L}_L = -\log p(Lev_t|\mathcal{C}_t, B_{t-1}). \tag{11}$$

The generated Lev_t is used for editing the B_{t-1} with the deterministic function f described in Equation 8.

The updated B_t is used to query the external knowledge (KB) and booking APIs. We first categorize the query result k_t according to the number of matching entities and the booking availability (a detailed list of k_t values is provided in the Appendix A). According to the result, we look up one embedding $e_k \in \mathbb{R}^{d_{model}}$ from the set of learnable KB state embeddings $E_k \in \mathbb{R}^{K \times d_{model}}$ [2], where K is the number of possible KB states. Then, the looked up embedding e_k is used as the starting token embedding of the response decoder for generating the delexicalized response R_t:

$$R_t = Decoder_R(H, e_k). \tag{12}$$

[2]KB state embeddings can be easily constructed by extending token embeddings of pre-trained models.

The learning objective of response generation is minimizing the negative log-likelihood of R_t given $B_{t-1}, \mathcal{C}_t$ and k_t,

$$\mathcal{L}_R = -\log p(R_t|\mathcal{C}_t, B_{t-1}, k_t). \tag{13}$$

Different from previous works (Lei et al., 2018; Zhang et al., 2019b), our response generation process is not condition on B_t because the dialogue context C_t already includes the information of B_t.

During training, all parameters are jointly optimized by minimizing the sum of the Lev generation and response generation losses:

$$\mathcal{L} = \mathcal{L}_L + \mathcal{L}_R. \tag{14}$$

3.3 Backbone Models

Our framework can be easily set up with pre-trained language models by initializing the encoder and decoders with pre-trained weights. We briefly introduce the two pre-trained backbones used in this paper: BART (Lewis et al., 2019) and Text-To-Text Transfer Transformer (T5) (Raffel et al., 2019).

BART is implemented as a standard encoder-decoder Transformer with a bidirectional encoder and an autoregressive decoder. It is pre-trained as denoising autoencoders which corrupt documents, and then optimize a reconstruction loss—the cross-entropy between the decoder's output and the original document. BART applies five different document corruption methods in the pre-training, including Token Masking (Devlin et al., 2019), Token Deletion, Text Infilling (Joshi et al., 2020), Sentence Permutation, and Document Rotation.

T5 is an encoder-decoder Transformer with relative position embeddings (Shaw et al., 2018). The model is pre-trained on the Colossal Clean Crawled Corpus (C4) (Raffel et al., 2019) that contains about 750GB of clean and natural English text. The pretraining objective is spans prediction, i.e., masking out 15% of input spans, then predicting the missing spans using the decoder.

4 Experiments

4.1 Datasets

We evaluate the proposed framework on the MultiWOZ dataset. It is a large-scale multi-domain task-oriented dialogue benchmark collected via the Wizard-of-Oz setting. The dataset contains 8438/1000/1000 dialogues for training/validation/testing, respectively. The dialogues

Model	Supervision			Inform (%)	Success (%)	BLEU	Combined
	Dialogue State	System Act	User Act				
Seq2Seq*	**oracle**	✗	✗	76.70	64.63	18.05	88.72
GPT2-small*	**oracle**	✗	✗	66.43	55.16	18.02	78.82
GPT2-medium*	**oracle**	✗	✗	70.96	61.36	19.05	85.21
MD-Sequicity	✓	✗	✗	75.72	58.32	15.40	82.40
HRED-TS*	✓	✓	✗	70.00	58.00	17.50	81.50
SFN + RL*	✓	✓	✗	73.80	58.60	18.27	84.47
DAMD	✓	✓	✗	72.79	60.43	16.93	83.54
DAMD + multi-action	✓	✓	✓	76.33	64.25	17.96	88.25
Sequicity (T5-small)	✓	✗	✗	71.64	61.01	18.02	84.35
MinTL (T5-small)	✓	✗	✗	80.04	72.71	**19.11**	95.49
MinTL (T5-base)	✓	✗	✗	82.15	74.44	18.59	96.88
MinTL (BART-large)	✓	✗	✗	**84.88**	**74.91**	17.89	**97.78**

Table 1: End-to-end response generation results on MultiWOZ2.0. ✓and ✗ denote whether a model leverages dialogue state, and/or speech act annotations during training. **oracle** denotes the gold dialogue state is used in both training and test time. Our results are averaged over three random seeds. *: results reported by the original paper.

in the corpus span over seven domains (restaurant, train, attraction, hotel, taxi, hospital, and police), and each dialogue session contains one to three domains. There are two existing dataset versions: MultiWOZ 2.0 (Budzianowski et al., 2018) and MultiWOZ 2.1 (Eric et al., 2019). We test the dialogue state tracking module of our framework on both datasets, and end-to-end models on MultiWOZ 2.0.

4.2 Implementation Details

We set up our framework with three pre-trained models: 1) T5-small (60M parameters) has 6 encoder-decoder layers and each layer has 8-headed attention with hidden size $d_{model} = 512$; 2) T5-base (220M parameters) has 12 encoder-decoder layers, and each of them has 12-headed attention with hidden size $d_{model} = 768$; 3) BART-large (400M parameters) has 12 encoder-decoder layers, each layer has 16-headed attention with hidden size $d_{model} = 1024$. We add special segment token embeddings and KB state embeddings to pre-trained models by extending the token embeddings. For a fair comparison, we use the pre-processing script released by Zhang et al. (2019b) [3]. All the models are fine-tuned with a batch size of 64 and early stop according to the performance on the validation set. Our implementation is based on HuggingFace Transformers library (Wolf et al., 2019a). We report the training hyper-parameters of each model in Appendix B.

[3]https://gitlab.com/ucdavisnlp/damd-multiwoz

4.3 Evaluation Metrics

For the end-to-end dialogue modeling task, there are three automatic metrics to evaluate the response quality: 1) **Inform** rate: if the system provides a correct entity, 2) **Success** rate: if the system provides the correct entity and answers all the requested information, 3) **BLEU** (Papineni et al., 2002) for measuring the fluency of the generated response. Following previous work (Mehri et al., 2019), we also report the combined score, i.e., **Combined** = (Inform + Success)$\times$0.5 + BLEU, as an overall quality measure. Joint goal accuracy (**Joint Acc.**) is used to evaluate the performance of the DST. The model outputs are only counted as correct when all of the predicted values exactly match the oracle values.

4.4 Baselines

4.4.1 End-to-end Modeling

Oracle DST: **Seq2Seq**, fine-tuned **GPT2-small**, and **GPT2-medium** (Radford et al., 2019) with oracle dialogue state as input (Budzianowski et al., 2018).

HRED-TS: a teacher-student framework with a hierarchical recurrent encoder-decoder backbone (Peng et al., 2019).

SFN + RL: a seq2seq network comprised of several pre-trained dialogue modules that are connected through hidden states. Reinforcement fine tuning is used additionally to train the model (Mehri et al., 2019).

Model	5%			10%			20%		
	Inform	Success	BLEU	Inform	Success	BLEU	Inform	Success	BLEU
MD-Sequicity	49.40	19.70	10.30	58.10	34.70	11.40	64.40	42.10	13.00
DAMD	57.20	27.00	9.90	58.30	33.90	13.30	67.40	40.10	13.80
DAMD + multi-action	56.60	24.50	10.60	62.00	39.40	14.50	68.30	42.90	11.80
MinTL (T5-small)	58.86	49.35	**14.51**	63.16	52.65	**15.71**	73.57	66.07	**17.55**
MinTL (T5-base)	69.57	57.76	14.50	72.17	61.16	15.56	78.98	**70.37**	16.69
MinTL (BART-large)	**75.48**	**60.96**	13.98	**78.08**	**66.87**	15.46	**82.48**	68.57	13.00

Table 2: Results of simulated low resource experiments. 5% (400 dialogues), 10% (800 dialogues), 20% (1600 dialogues) of training data is used to train each model.

Model	Inform (%)	Success (%)	BLEU
MinTL (T5-small)	**80.04**	**72.71**	19.11
w/o Lev	71.62	63.20	16.11
w/ shared decoder	74.90	67.03	**20.10**

Table 3: Ablation study on different variants of *MinTL* on MultiWOZ 2.0 in the end-to-end evaluation setting.

MD-Sequicity: an extension of the *Sequicity* (Lei et al., 2018) framework for multi-domain task-oriented dialogue by Zhang et al. (2019b).

DAMD: the domain-aware multi-decoder network proposed by Zhang et al. (2019b). The author also proposed the multi-action data augmentation method by leveraging system act and user act annotations. We denote the method as **DAMD + multi-action**.

Sequicity + T5: The *Sequicity* (Lei et al., 2018) framework with the T5 backbone model (Raffel et al., 2019). There are two main differences between *Sequicity* and our framework: 1) *Sequicity* generates dialogue states from scratch at each turn, 2) *MinTL* generates responses by conditioning on dialogue context C_t instead of new generated dialogue state B_t.

4.4.2 Dialogue State Tracking

We compare our DST module with both the classification-based DST and generation-based DST baselines. The former includes MDBT (Ramadan et al., 2018), GLAD (Zhong et al., 2018), GCE (Nouri and Hosseini, 2018), FJST (Eric et al., 2019), HyST (Goel et al., 2019), SUMBT (Lee et al., 2019), SST (Chen et al., 2020), TOD-BERT (Wu et al., 2020), and DST-Picklist (Zhang et al., 2019a); the latter includes Neural Reading (Gao et al., 2019), TRADE (Wu et al., 2019a), COMER (Ren et al., 2019), SOM-DST (Kim et al., 2019), DSTQA (Zhou and Small, 2019), and NADST (Le et al., 2020).

4.5 Results

4.5.1 End-to-end Modeling

We first compare our systems with baselines in the end-to-end dialogue learning setting, where the generated dialogue states are used for the knowledge base search and response generation. The results are shown in Table 1. *MinTL*-based systems achieve the best performance in terms of inform rate, success rate, and BLEU. With fewer human annotations, our models improve the previous SOTA model (Zhang et al., 2019b) by around a 10% success rate. Using T5-small as the backbone barely improves the overall performance of *Sequicity* (Lei et al., 2018), because the copy mechanism (Gu et al., 2016) is absent in this pre-trained model. Compared to the *Sequicity* framework, our approach achieves an around 11% higher success rate with the same backbone model, which suggests that *MinTL* is able to effectively leverage pre-trained language models.

Low Resource Settings. We evaluate our models in the simulated low resource setting to test if transferring a pre-trained language model to task-oriented dialogue can alleviate the data scarcity problem. Specifically, we use 5%, 10%, and 20% of the training set data to train our models and baselines. The result is reported in Table 2. *MinTL*-based systems consistently outperform the DAMD (Zhang et al., 2019b), MD-Sequicity (Lei et al., 2018) baselines by a large margin, which demonstrates the effectiveness of transfer learning. It is worth noting that the performance gap between *MinTL* and baselines decreases with respect to the increase in the training data size. This indicates that prior knowledge from the pre-trained language model is more important in the extremely low-resource scenarios. With only 20% of training

Model	MWoZ Joint Acc.	
	2.0	**2.1**
MDBT (Ramadan et al., 2018)†	15.57	-
GLAD (Zhong et al., 2018)†	35.57	-
GCE (Nouri and Hosseini, 2018)†	36.27	-
FJST (Eric et al., 2019)⋆	40.20	38.00
HyST (Goel et al., 2019)†	44.24	-
SUMBT (Lee et al., 2019)†	46.65	-
TOD-BERT (Wu et al., 2020)⋆	-	48.00
DST-Picklist (Zhang et al., 2019a)⋆	-	53.30
SST (Chen et al., 2020)⋆	**51.17**	**55.23**
Neural Reading (Gao et al., 2019)†	41.10	-
TRADE (Wu et al., 2019a)†	48.62	45.60
COMER (Ren et al., 2019)†	48.79	-
DSTQA (Zhou and Small, 2019)†	51.44	51.17
SOM-DST (Kim et al., 2019)⋆	51.38	52.57
NADST (Le et al., 2020)⋆	50.52	49.04
MinTL (T5-small)	51.24	50.95
MinTL (T5-base)	52.07	52.52
MinTL (BART-large)	**52.10**	**53.62**

Table 4: Dialogue state tracking results on MultiWOZ 2.0 and MultiWOZ 2.1. The upper part and lower part of the table show the joint goal accuracy of the classification-based and generation-based model, respectively. †: results reported by the leaderboard. ⋆: results reported by the original paper.

data, our models can achieve competitive results compared to the full data trained DAMD model.

Ablation Study. We conduct a simple ablation study with the T5-small backbone to understand the different variants of *MinTL*. We test our framework with: 1) the belief span proposed by Lei et al. (2018), and 2) sharing the decoder parameter for both Lev generation and response generation. The result is reported in Table 3. Replacing Lev with *belief span* hurts the overall performance, which shows the effectiveness of Lev. In section 4.5.2, we also show that Lev greatly reduces the inference latency. On the other hand, although the Lev generation and response generation are conditioned on different starting tokens, sharing the parameters of the two decoders decreases both inform and success rate. It is important to decouple the two decoders because the distributions between the Lev decoder and response decoder are different.

4.5.2 Dialogue State Tracking

Table 4 reports the DST results on MultiWOZ 2.0 and MultiWOZ 2.1. *MinTL*-based BART model achieves the highest joint goal accuracy among the generation-based DST models on both datasets. Compared to the SOTA classification-based DST model SST (Chen et al., 2020), our model obtains a 1.62% lower joint goal accuracy on MultiWOZ 2.1. This is because classification-based models have the advantage of predicting slot values from valid candidates. However, having one classifier per domain-slot pair is not scalable when the number of slots and values grow (Lei et al., 2018). In contrast, our model only generates minimal slot-value pairs when necessary. In our error analysis, we found that our model sometimes generates invalid slot values (e.g., *the cambridge punte* instead of *the cambridge punter* for the *taxi-destination* slot), which can be avoided with a full ontology constraint.

Model	Joint Acc	Latency	Speed Up	NoT
TRADE⋆	45.60	362.15	×2.12	-
TSCP⋆	37.12	767.57	×1.00	-
NADST⋆	49.04	**27.31**	×28.11	-
Sequicity (T5-small)	44.10	200.48	×3.83	20.99
MinTL (T5-small)	**50.95**	49.26	×15.58	6.58

Table 5: Latency analysis on MultiWOZ 2.1. **Latency** denotes the average inference time (ms) per turn and **NoT** denotes the average number of generated tokens per turn. ⋆: results borrowed from Le et al. (2020)

Latency Analysis. Table 5 reports the average inference time (ms) of each model on the test set of MultiWOZ 2.1. Following Le et al. (2020), we compute the latency of each model on Nvidia V100 with a batch size of 1. Our model is 15 times faster than TSCP (Lei et al., 2018) and around 7 times faster than TRADE (Wu et al., 2019a). On the other hand, our model is slower than NADST (Le et al., 2020), which is explicitly optimized for inference speed using the non-autoregressive decoding strategy. However, it is hard to incorporate NADST into end-to-end response generation models due to its task-specific architecture design (e.g., fertility decoder). Finally, we compare the generative DST modules of two end-to-end models. By using same backbone model, *MinTL* is around 4 times faster than Sequicity by generating only 6 tokens per turn, which suggests that Lev significantly improves the inference efficiency.

5 Conclusion

In this paper, we proposed *MinTL*, a simple and general transfer learning framework that effectively leverages pre-trained language models to jointly learn DST and dialogue response generation. The Lev is proposed for reducing the DST complexity and improving inference efficiency. In addition, two pre-trained Seq2Seq language models: T5 (Raffel et al., 2019) and BART (Lewis et al., 2019) are incorporated in our framework. Experimental results on MultiWOZ shows that, by using *MinTL*, our systems not only achieve new SOTA result on both dialogue state tracking and end-to-end response generation but also improves the inference efficiency. In future work, we plan to explore task-oriented dialogues domain-adaptive pre-training methods (Wu et al., 2020; Peng et al., 2020) to enhance our language model backbones, and extend the framework for mixed chit-chat and task-oriented dialogue agents (Madotto et al., 2020a).

Acknowledgements

This work has been partially funded by MRP/055/18 of the Innovation Technology Commission, The Hong Kong SAR Government. We would like to thanks Yichi Zhang for helpful discussion.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's gpt-2-how can i help you? towards the use of pre-trained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22.

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.

Lu Chen, Boer Lv, Chunxin Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *AAAI 2020*.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067.

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.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

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*.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49. Association for Computational Linguistics.

Mihail Eric and Christopher D Manning. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. *arXiv preprint arXiv:1701.04024*.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, Dilek Hakkani-Tur, and Amazon Alexa AI. 2019. 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*, pages 1458–1462.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. 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)*, pages 1631–1640.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert:

Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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

Hung Le, Richard Socher, and Steven C.H. Hoi. 2020. 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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2020. Exploring versatile generative language model via parameter-efficient transfer learning. *arXiv preprint arXiv:2004.03829*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrea Madotto, Zhaojiang Lin, Yejin Bang, and Pascale Fung. 2020a. The adapter-bot: All-in-one controllable conversational model. *arXiv preprint arXiv:2008.12579*.

Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, Jamin Shin, and Pascale Fung. 2020b. Attention over parameters for dialogue systems. *arXiv preprint arXiv:2001.01871*.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *arXiv preprint arXiv:1804.08217*.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177.

Arvind Neelakantan, Semih Yavuz, Sharan Narang, Vishaal Prasad, Ben Goodrich, Daniel Duckworth, Chinnadhurai Sankar, and Xifeng Yan. 2019. Neural assistant: Joint action prediction, response generation, and latent knowledge reasoning. *arXiv preprint arXiv:1910.14613*.

Elnaz Nouri and Ehsan Hosseini. 2018. Toward scalable neural dialogue state tracking model. *arXiv preprint arXiv:1812.00899*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Shuke Peng, Xinjing Huang, Zehao Lin, Feng Ji, Haiqing Chen, and Yin Zhang. 2019. Teacher-student framework enhanced multi-domain dialogue generation. *arXiv preprint arXiv:1908.07137*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 432–437.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

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.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019a. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019b. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogues. *arXiv preprint arXiv:2004.06871*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019a. 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.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2019b. Alternating recurrent dialog model with large-scale pre-trained language models. *arXiv preprint arXiv:1910.03756*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

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. 2019a. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2019b. Task-oriented dialog systems that consider multiple appropriate responses under the same context. *arXiv preprint arXiv:1911.10484*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019c. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

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*.

A Appendices

A.1 KB States

Table 6 shows KB states that are categorized by the number of matching entities and booking availability. T_1, T_2 are thresholds of the number of match entities. We define $T_1 = 1$ and $T_2 = 3$ for train domain, $T_1 = 5$ and $T_2 = 10$ for other domains.

KB States	Entity Match	Book Availability
KB1	-	-
KB2	0	-
KB3	$\leq T_1$	-
KB4	$\leq T_2$	-
KB5	$> T_2$	-
KB6	-	fail
KB7	0	fail
KB8	$\leq T_1$	fail
KB9	$\leq T_2$	fail
KB10	$> T_2$	fail
KB11	-	success
KB12	0	success
KB13	$\leq T_1$	success
KB14	$\leq T_2$	success
KB15	$> T_2$	success

Table 6: KB states categorized by the number of matching entities and booking availability. T_1 and T_2 are thresholds. We define $T_1 = 1$ and $T_2 = 3$ for train domain, $T_1 = 5$ and $T_2 = 10$ for other domains.

A.2 Hyper-parameters

We report our training hyper-parameters on each task, which includes context window size `w`, learning rate `lr`, and learning rate decay rate `lr-decay`. We decay the learning rate when the performance in validation set does not improve. All of models are trained on Nvidia V100.

Task	Model	w	lr	lr-decay
End-to-End Response Generation	T5-small	2	6e-4	0.8
	T5-base	2	6e-4	0.8
	BART-large	2	3e-5	0.8
DST	T5-small	3	6e-4	0.8
	T5-base	3	6e-4	0.8
	BART-large	3	1e-5	0.8

Table 7: This table reports the hyper-parameters used by each model. w: window size, `lr`: learning rate, `lr-decay`: learning rate decay rate per epoch.

A.3 Generated Examples

Table 8 - 10 shows the end-to-end response generation examples by *MinTL* with T5-small, T5-base, and BART-large backbones.

MUL0113		Dialogue
Turn 1	User	i am in need of an expensive indian restaurant for dinner .
	Dialogue State	[restaurant] food indian pricerange expensive
	Delex. Response	there are [value_choice] [value_price] [value_food] restaurants . do you have a preference for area ?
	Lex. Response	There are 14 expensive indian restaurants . Do you have a preference for area ?
	Gold Response	there are 14 such restaurants . do you want to be in a specific part of town ?
Turn 2	User	i don't care what part of town it is in , i would like a phone number please .
	Dialogue State	[restaurant] food indian pricerange expensive
	Delex. Response	[value_name] is in the [value_area] and is in the [value_price] price range . their phone number is [value_phone] .
	Lex. Response	Curry Garden is in the centre and is in the expensive price range . Their phone number is 01223302330 .
	Gold response	sure , i am sorry . i have curry garden their phone number is 01223302330 . would you like me to reserve a table for you ?
Turn 3	User	not at this time . i am looking for a 3-star hotel , though . no guest house .
	Dialogue State	[restaurant] food indian pricerange expensive [hotel] stars 3 type hotel
	Delex. Response	there are [value_choice] [value_type] that meet your criteria . [value_name] and [value_name] . would you like me to book 1 of these for you ?
	Lex. Response	There are 2 hotel that meet your criteria . Gonville Hotel and the Lensfield Hotel . Would you like me to book 1 of these for you ?
	Gold response	i have found the gonville hotel . it is a 3 star hotel . would you like me to book it for you ?
Turn 4	User	not yet . is the gonville hotel in the same price range as the curry garden ?
	Dialogue State	[restaurant] food indian pricerange expensive [hotel] stars 3 type hotel
	Delex. Response	yes , it is in the [value_price] price range . would you like me to book it for you ?
	Lex. Response	yes , it is in the expensive price range . would you like me to book it for you ?
	Gold response	yes , it is an expensive hotel .
Turn 5	User	can you book a room for 2 people for 2 nights starting on sunday ?
	Dialogue State	[restaurant] food indian pricerange expensive [hotel] stars 3 type hotel people 2 day sunday stay 2
	Delex. Response	I am sorry , but the booking was unsuccessful . Would you like to try another day or a shorter stay ?
	Lex. Response	I am sorry , but the booking was unsuccessful . Would you like to try another day or a shorter stay ?
	Gold response	the booking was unsuccessful . would you like to try for another day or a shorter stay ?
Turn 6	User	could you try it for 1 night ? if that works , please book it and get the reference number .
	Dialogue State	[restaurant] food indian pricerange expensive [hotel] stars 3 type hotel people 2 day sunday stay 1
	Delex. Response	booking was successful . reference number is : [value_reference] .
	Lex. Response	Booking was successful . Reference number is : 5VRJSMSE .
	Gold response	you are all set for 1 night (sunday) , reference number is 6wvv053q . would you like to try another location for monday night ?
Turn 7	User	no . that is all i need , thank you .
	Dialogue State	[restaurant] food indian pricerange expensive [hotel] stars 3 type hotel people 2 day sunday stay 1
	Delex. Response	You are welcome . Have a great day .
	Lex. Response	You are welcome . Have a great day .
	Gold response	thank you , and goodbye .

Table 8: *MinTL* (T5-small) end-to-end response generation on the test set of MultiWOZ. We show the dialogue states, delexicalized responses, lexicalized responses generated by our model.

PMUL2000		Dialogue
Turn 1	User	i am looking for information on a hotel called hobsons house .
	Dialogue State	[hotel] name hobsons house
	Delex. Response	[value_name] is a [value_type] in the [value_area] . it is [value_price] and has a [value_stars] star rating . would you like to book a room ?
	Lex. Response	Hobsons House is a guest house in the west . It is moderate and has a 3 star rating . Would you like to book a room ?
	Gold response	hobsons house is a guest house found in the west side of town . it is in the moderate price range and is rated 3 stars .
Turn 2	User	do they have free parking ?
	Dialogue State	[hotel] name hobsons house
	Delex. Response	yes , they do have free parking .
	Lex. Response	Yes , they do have free parking .
	Gold response	yes hobsons house has both free internet and parking . would you like to book a room ?
Turn 3	User	can i please get their postcode ?
	Dialogue State	[hotel] name hobsons house
	Delex. Response	the postcode is [value_postcode] . is there anything else i can help you with today?
	Lex. Response	The postcode is CB39LH . Is there anything else i can help you with today ?
	Gold response	yes , of course ! it is cb39lh .
Turn 4	User	i also need a train that should depart from kings lynn and should go to cambridge .
	Dialogue State	[hotel] name hobsons house [train] destination cambridge departure kings lynn
	Delex. Response	there are [value_choice] trains that leave from [value_departure] to [value_destination] . what day would you like to travel ?
	Lex. Response	There are 133 trains that leave from kings lynn to cambridge . What day would you like to travel ?
	Gold response	what day will you travel ?
Turn 5	User	i 'll be traveling on monday and need to arrive by 10 :00 .
	Dialogue State	[hotel] name hobsons house [train] destination cambridge departure kings lynn day monday arrive 10 :00
	Delex. Response	the [value_id] leaves at [value_leave] and arrives at [value_arrive] . would you like me to book that for you ?
	Lex. Response	The TR4943 leaves at 05:11 and arrives at 05:58 . Would you like me to book that for you ?
	Gold response	the tr0942 leaves kings lynn at 09 :11 and arrives in cambridge at 09 :58 . would you like to book a reservation ?
Turn 6	User	yes can i book it for 8 people ?
	Dialogue State	[hotel] name hobsons house [train] destination cambridge departure kings lynn day monday arrive 10 :00 people 8
	Delex. Response	booking was successful , the total fee is [value_price] payable at the station . reference number is : [value_reference] .
	Lex. Response	Booking was successful , the total fee is 9.80 pounds payable at the station . Reference number is : I4AFI8ET .
	Gold response	you are all set ! your reference # is 2pffj67j . total cost is 78 .40 gbp , payable at the station . can i help you with anything else today ?
Turn 7	User	no , thank you . i have everything i need .
	Dialogue State	[hotel] name hobsons house [train] destination cambridge departure kings lynn day monday arrive 10 :00 people 8
	Delex. Response	you are welcome . have a great day .
	Lex. Response	You are welcome . Have a great day .
	Gold response	you have a wonderful day . goodbye .

Table 9: *MinTL* (T5-base) end-to-end response generation on the test set of MultiWOZ. We show the dialogue states, delexicalized responses, lexicalized responses generated by our model.

MUL0570		Dialogue
Turn 1	User	hello , do you know of any points of interest in the centre of cambridge ?
	Dialogue state:	[attraction] area centre
	Delexicalized	there are [value_choice] attractions in the [value_area] . what type of attraction are you interested in ?
	lexicalized	There are 44 attractions in the centre . What type of attraction are you interested in ?
	Gold response	there are 44 attractions in the center of town . what types are you interested in ?
Turn 2	User	i am not sure . what types of attractions are there ?
	Dialogue state:	[attraction] area centre
	Delexicalized	there are [value_type] , [value_type] , and [value_type] .
	Lexicalized	There are theatre , architecture , and nightclub .
	Gold response	you could visit a church or college . there are also 8 museums that are free of charge to enter .
Turn 3	User	i would like to visit a college . may i have the entrance fee and postcard ?
	Dialogue state:	[attraction] area centre type college
	Delexicalized	there are [value_choice] [value_type] in the [value_area] . [value_name] is [value_price] to enter .
	Lexicalized	There are 13 college in the centre . Christ 's College is free to enter .
	Gold response	there is christ 's college , which has a free entrance fee . their postcode is cb23bu .
Turn 4	User	can i have information for a train departing from ely ? i am looking to arrive in cambridge by about 15 :15 on tuesday .
	Dialogue state:	[attraction] area centre type college [train] destination cambridge day tuesday arrive 15 :15 departure ely
	Delexicalized	there are [value_choice] trains that fit your needs . what time would you like to leave ?
	Lexicalized	There are 5 trains that fit your needs . What time would you like to leave ?
	Gold response	tr3412 will arrive around 11 :52 on tuesday if you are interested in that .
Turn 5	User	i 'll take it for just me what is the travel time , and the price please ?
	Dialogue state:	[attraction] area centre type college [train] destination cambridge day tuesday arrive 15 :15 departure ely
	Delexicalized	the travel time is [value_time] and the price is [value_price] .
	Lexicalized	The travel time is 17 minutes and the price is 4.40 pounds .
	Gold response	the travel time is 17 minutes and the price is 4 .40 gbp .
Turn 6	User	great thank you i think i have all the information i need
	Dialogue state:	[attraction] area centre type college [train] destination cambridge day tuesday arrive 15 :15 departure ely
	Delexicalized	Thank you for using our service today !
	Lexicalized	Thank you for using our service today !
	Gold response	would you like me to book that train for you ?
Turn 7	User	no thank you . appreciate the help !
	Dialogue state:	[attraction] area centre type college [train] destination cambridge day tuesday arrive 15 :15 departure ely
	Delexicalized	thank you for using our service . have a great day . goodbye .
	Lexicalized	thank you for using our service . have a great day . goodbye .
	Gold response	you are welcome . have a good day !

Table 10: *MinTL* (BART-large) end-to-end response generation on the test set of MultiWOZ. We show the dialogue states, delexicalized responses, lexicalized responses generated by our model.