A Sequence-to-Sequence Approach to Dialogue State Tracking

Yue Feng[†][*]   **Yang Wang**[‡]   **Hang Li**[‡]
†University College London, London, UK
‡ByteDance AI Lab, Beijing, China
† yue.feng.20@ucl.ac.uk
‡{wangyang.127, lihang.lh}@bytedance.com

Abstract

This paper is concerned with dialogue state tracking (DST) in a task-oriented dialogue system. Building a DST module that is highly effective is still a challenging issue, although significant progresses have been made recently. This paper proposes a new approach to dialogue state tracking, referred to as Seq2Seq-DU, which formalizes DST as a sequence-to-sequence problem. Seq2Seq-DU employs two BERT-based encoders to respectively encode the utterances in the dialogue and the descriptions of schemas, an attender to calculate attentions between the utterance embeddings and the schema embeddings, and a decoder to generate pointers to represent the current state of dialogue. Seq2Seq-DU has the following advantages. It can jointly model intents, slots, and slot values; it can leverage the rich representations of utterances and schemas based on BERT; it can effectively deal with categorical and non-categorical slots, and unseen schemas. In addition, Seq2Seq-DU can also be used in the NLU (natural language understanding) module of a dialogue system. Experimental results on benchmark datasets in different settings (SGD, MultiWOZ2.2, MultiWOZ2.1, WOZ2.0, DSTC2, M2M, SNIPS, and ATIS) show that Seq2Seq-DU outperforms the existing methods.

1 Introduction

A task-oriented dialogue system usually consists of several modules: natural language understanding (NLU), dialogue state tracking (DST), dialogue policy (Policy), and natural language generation (NLG). We consider DST and also NLU in this paper. In NLU, a semantic frame representing the content of user utterance is created in each turn

---

The work was done when the first author was an intern at ByteDance AI Lab.
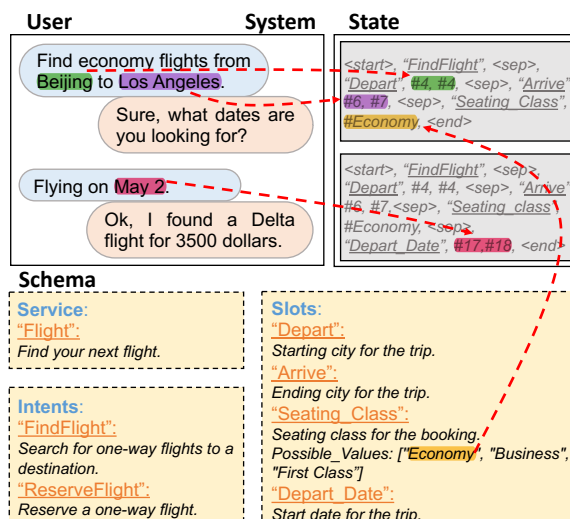

Figure 1: An example of dialogue state tracking. Given a dialogue history that contains user utterances and system utterances, and descriptions of schema that contain all possible intents and slot-value pairs, a dialogue state for the current turn is created which is represented by intents and slot-value pairs. There are slot values obtained from the schema (categorical) as well as slot values extracted from the utterances (non-categorical). #4, #6, etc denote pointers.

of dialogue. In DST, several semantic frames representing the 'states' of dialogue are created and updated in multiple turns of dialogue. Domain knowledge in dialogues is represented by a representation referred to as schema, which consists of possible intents, slots, and slot values. Slot values can be in a pre-defined set, with the corresponding slot being referred to as categorical slot, and they can also be from an open set, with the corresponding slot being referred to as non-categorical slot. Figure 1 shows an example of DST.

We think that a DST module (and an NLU module) should have the following abilities. (1) Global, the model can jointly represent intents, slots, and slot values. (2) Represenable, it has strong capa-

bility to represent knowledge for the task, on top of a pre-trained language model like BERT. (3) Scalable, the model can deal with categorical and non-categorical slots and unseen schemas.

Many methods have been proposed for DST (Wu et al., 2019; Zhong et al., 2018; Mrkšić et al., 2017; Goo et al., 2018). There are two lines of relevant research. (1) To enhance the scalability of DST, a problem formulation, referred to as schema-guided dialogue, is proposed. In the setting, it is assumed that descriptions on schemas in natural language across multiple domains are given and utilized. Consequently, a number of methods are developed to make use of schema descriptions to increase the scalability of DST (Rastogi et al., 2019; Zang et al., 2020; Noroozi et al., 2020). The methods regard DST as a classification and/or an extraction problem and independently infer the intent and slot value pairs for the current turn. Therefore, the proposed models are generally representable and scalable, but not global. (2) There are also a few methods which view DST as a sequence to sequence problem. Some methods sequentially infer the intent and slot value pairs for the current turn on the basis of dialogue history and usually employ a hierarchical structure (not based on BERT) for the inference (Lei et al., 2018; Ren et al., 2019; Chen et al., 2020b). Recently, a new approach is proposed which formalizes the tasks in dialogue as sequence prediction problems using a unified language model (based on GPT-2) (Hosseini-Asl et al., 2020). The method cannot deal with unseen schemas and intents, however, and thus is not scalable.

We propose a novel approach to DST, referred to as Seq2Seq-DU (sequence-to-sequence for dialogue understanding), which combines the advantages of the existing approaches. To the best of our knowledge, there was no previous work which studied the approach. We think that DST should be formalized as a sequence to sequence or 'translation' problem in which the utterances in the dialogue are transformed into semantic frames. In this way, the intents, slots, and slot values can be jointly modeled. Moreover, NLU can also be viewed as a special case of DST and thus Seq2Seq-DU can also be applied to NLU. We note that very recently the effectiveness of the sequence to sequence approach has also been verified in other language understanding tasks (Paolini et al., 2021).

Seq2Seq-DU comprises a BERT-based encoder to encode the utterances in the dialogue, a BERT based encoder to encode the schema descriptions, an attender to calculate attentions between the utterance embeddings and schema embeddings, and a decoder to generate pointers of items representing the intents and slots-value pairs of state.

Seq2Seq-DU has the following advantages. (1) Global: it relies on the sequence to sequence framework to simultaneously model the intents, slots, and slot-values. (2) Representable: It employs BERT (Devlin et al., 2019) to learn and utilize better representations of not only the current utterance but also the previous utterances in the dialogue. If schema descriptions are available, it also employs BERT for the learning and utilization of their representations. (3) Scalable: It uses the pointer generation mechanism, as in the Pointer Network (Vinyals et al., 2015), to create representations of intents, slots, and slot-values, no matter whether the slots are categorical or non-categorical, and whether the schemas are unseen or not.

Experimental results on benchmark datasets show that Seq2Seq-DU[1] performs much better than the baselines on SGD, MultiWOZ2.2, and Multi-WOZ2.1 in multi-turn dialogue with schema descriptions, is superior to BERT-DST on WOZ2.0, DSTC2, and M2M, in multi-turn dialogue without schema descriptions, and works equally well as Joint BERT on ATIS and SNIPS in single turn dialogue (in fact, it degenerates to Joint BERT).

2 Related Work

There has been a large amount of work on task-oriented dialogue, especially dialogue state tracking and natural language understanding (eg., (Zhang et al., 2020; Huang et al., 2020; Chen et al., 2017)). Table 1 makes a summary of existing methods on DST. We also indicate the methods on which we make comparison in our experiments.

2.1 Dialogue State Tracking

Previous approaches mainly focus on encoding of the dialogue context and employ deep neural networks such as CNN, RNN, and LSTM-RNN to independently infer the values of slots in DST (Mrkšić et al., 2017; Xu and Hu, 2018; Zhong et al., 2018; Ren et al., 2018; Rastogi et al., 2017; Ramadan et al., 2018; Wu et al., 2019; Zhang et al., 2019; Heck et al., 2020). The approaches

---

[1]The code is available at `https://github.com/sweetalyssum/Seq2Seq-DU`.

Model	Characteristics	Data Sets	Comparison
FastSGD (Noroozi et al., 2020)	BERT-based model, employs two carry-over procedures and multi-head attentions to model schema descriptions.	SGD	Yes
SGD Baseline (Rastogi et al., 2019)	BERT-based model, predictions are made over a dynamic set of intents and slots, using their descriptions.	SGD and MultiWOZ2.2	Yes
TripPy (Heck et al., 2020)	BERT-based model, make use of various copy mechanisms to fill slots with values.	MultiWOZ2.2	Yes
TRADE (Wu et al., 2019)	Generate dialogue states from utterances using a copy mechanism, facilitating knowledge transfer for new schema elements.	MultiWOZ2.2	Yes
DS-DST (Zhang et al., 2019)	BERT-based model, to classify over a candidate list or find values from text spans.	MultiWOZ2.2	Yes
BERT-DST (Chao and Lane, 2019)	Use BERT as dialogue context encoder and makes parameter sharing across slots.	DSTC2, WOZ2.0, and M2M	Yes
StateNet (Ren et al., 2018)	Independent of number of values, shares parameters across slots and uses pre-trained word vectors.	DSTC2 and WOZ2.0	Yes
GLAD (Zhong et al., 2018)	Use global modules to share parameters across slots and uses local modules to retrain slot-specific parameters.	DSTC2 and WOZ2.0	Yes
Belief Tracking (Ramadan et al., 2018)	Utilize semantic similarity between dialogue utterances and ontology, and information is shared across domains.	DSTC2 and WOZ2.0	Yes
Neural Belief Tracker (Mrkšić et al., 2017)	Conduct reasoning on pre-trained word vectors, and combines them into representations of user utterance and dialogue context.	DSTC2 and WOZ2.0	Yes
DST+LU (Rastogi et al., 2018)	Select candidates for each slot, while candidates are generated by NLU.	M2M	Yes
Joint BERT (Chen et al., 2019)	A joint intent classification and slot filling model based on BERT.	ATIS and SNIPS	Yes
Slot-Gated (Goo et al., 2018)	Use a slot gate, models relation between intent and slot vectors to create semantic frames.	ATIS and SNIPS	Yes
Atten.-BiRNN (Liu and Lane, 2016)	Attention-based model, explores several strategies for alignment between intent classification and slot labeling.	ATIS and SNIPS	Yes
RNN-LSTM (Hakkani-Tür et al., 2016)	Use RNN with LSTM cells to create complete semantic frames from user utterances.	ATIS and SNIPS	Yes
Sequicity (Lei et al., 2018)	Two-stage sequence-to-sequence model based on CopyNet, conducts both dialogue state tracking and response generation.	CamRest676 and KVRET	No
COMER (Ren et al., 2019)	BERT-based hierarchical encoder-decoder model, generates state sequence based on user utterance	WOZ2.0 and MultiWOZ2.0	Yes
CREDIT (Chen et al., 2020b)	Hierarchical encoder-decoder model, views DST as a sequence generation problem.	MultiWOZ2.0 and MultiWOZ2.1	No
SimpleTOD (Hosseini-Asl et al., 2020)	A unified sequence-to-sequence model based on GPT-2, conducts dialogue state tracking, dialogue action prediction, and response generation.	MultiWOZ2.0 and MultiWOZ2.1	Yes

Table 1: Summary of existing methods on DST.

cannot deal with unseen schemas in new domains, however. To cope with the problem, a new direction called schema-guided dialogue is proposed recently, which assumes that natural language descriptions of schemas are provided and can be used to help transfer knowledge across domains. As such, a number of methods are developed in the recent dialogue competition SGD (Rastogi et al., 2019; Zang et al., 2020; Noroozi et al., 2020; Chen et al., 2020a). Our work is partially motivated by the SGD initiative. Our model Seq2Seq-DU is unique in that it formalizes schema-guided DST as a sequence-to-sequence problem using BERT and pointer generation.

In fact, sequence-to-sequence models are also utilized in DST. Sequicity (Lei et al., 2018) is a two-step sequence to sequence model which first encodes the dialogue history and generates a belief span, and then generates a language response from the belief span. COMER (Ren et al., 2019) and CREDIT (Chen et al., 2020b) are hierarchical sequence-to-sequence models which represent the intents and slot-value pairs in a hierarchical way, and employ a multi-stage decoder. Simple-TOD (Hosseini-Asl et al., 2020) is a unified approach to task-oriented dialogue which employs

a single and causal language model to perform sequence prediction in DST, Policy, and NLG. Our proposed approach also uses a sequence-to-sequence model. There are significant differences between our model Seq2Seq-DU and the existing models. First, there is no hierarchy in decoding of Seq2Seq-DU. A flat structure on top of BERT appears to be sufficient for jointly capturing the intents, slots, and values. Second, the decoder in Seq2Seq-DU generates pointers instead of tokens, and thus can easily and effectively handle categorical slots, non-categorical slots, as well as unseen schemas.

2.2 Natural Language Understanding

Traditionally the problem of NLU is decomposed into two independent issues, namely classification of intents and sequence labeling of slot-value pairs (Liu and Lane, 2016; Hakkani-Tür et al., 2016). For example, deep neural network combined with conditional random field is employed for the task (Yao et al., 2014). Recently the pre-trained language model BERT (Chen et al., 2019) is exploited to further enhance the accuracy. Methods are also proposed which can jointly train and utilize classification and sequence labeling models (Chen
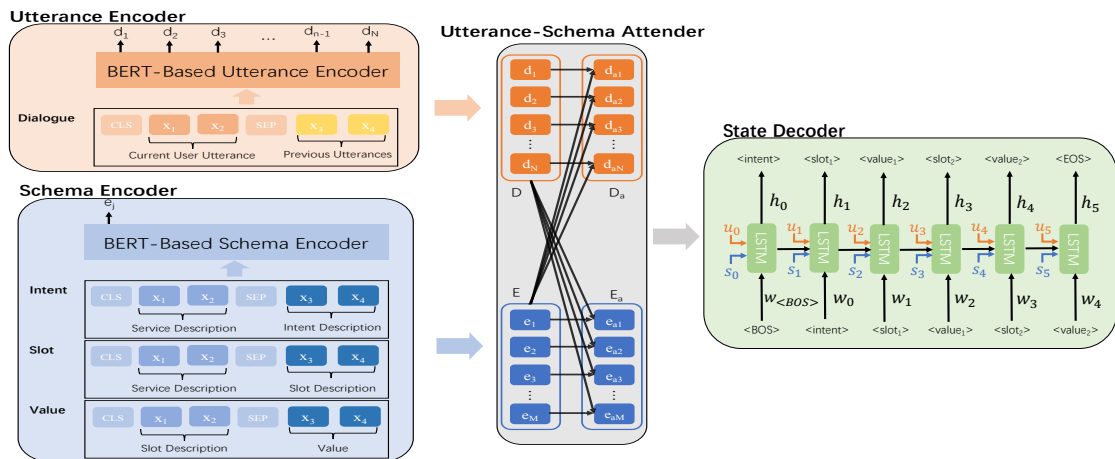
Figure 2: The architecture of Seq2Seq-DU, containing utterance encoder, schema encoder, utterance-schema attender, and state decoder.

et al., 2019; Goo et al., 2018). In this paper, we view NLU as special case of DST and employ our model Seq2Seq-DU to perform NLU. Seq2Seq-DU can degenerate to a BERT based NLU model.

3 Our Approach

Our approach Seq2Seq-DU formalizes dialogue state tracking as a sequence to sequence problem using BERT and pointer generation. As shown in Figure 2, Seq2Seq-DU consists of an utterance encoder, a schema encoder, an utterance schema attender, and a state decoder. In each turn of dialogue, the utterance encoder transforms the current user utterance and the previous utterances in the dialogue into a sequence of utterance embeddings using BERT; the schema encoder transforms the schema descriptions into a set of schema embeddings also using BERT; the utterance schema attender calculates attentions between the utterance embeddings and the schema embeddings to create attended utterance and schema representations; finally, the state decoder sequentially generates a state representation on the basis of the attended representations using LSTM and pointer generation.

3.1 Utterance Encoder

The utterance encoder takes the current user utterance as well as the previous utterances (user and system utterances) in the dialogue (a sequence of tokens) as input and employs BERT to construct a sequence of utterance embeddings. The relations between the current utterance and the previous utterances are captured by the encoder.

The input of the encoder is a sequence of tokens with length N, denoted as $X = (x_1, ..., x_N)$. The first token x_1 is [CLS], followed by the tokens of the current user utterance and the tokens of the previous utterances, separated by [SEP]. The output is a sequence of embeddings also with length N, denoted as $D = (d_1, ..., d_N)$ and referred to as utterance embeddings, with one embedding for each token.

3.2 Schema Encoder

The schema encoder takes the descriptions of intents, slots, and categorical slot values (a set of combined sequences of tokens) as input and employs BERT to construct a set of schema embeddings.

Schema	Sequence 1	Sequence 2
Intent	service description	intent description
Slot	service description	slot description
Value	slot description	value

Table 2: Descriptions for a dialogue schema. Two combined descriptions are used for describing an intent, a slot, or a value in the schema.

Suppose that there are I intents, S slots, and V categorical slot values in the schemas. Each schema element is described by two descriptions as outlined in Table 2. The input is a set of combined sequences of tokens, denoted as $Y = \{y_1, ..., y_M\}$. Note that $M = I + S + V$. Each combined sequence starts with [CLS], followed by the tokens of the two descriptions with [SEP] as a separator. The final representation of [CLS] is used as the embedding of the input intent, slot, or slot value. The output is a set of embeddings, and all the embeddings are called schema embeddings $E = \{e_1, ..., e_M\}$.

The schema encoder in fact adopts the same approach of schema encoding as in (Rastogi et al., 2019). There are two advantages with the approach. First, the encoder can be trained across different domains. Schema descriptions in different domains can be utilized together. Second, once the encoder is fine-tuned, it can be used to process unseen schemas with new intents, slots, and slot values.

3.3 Utterance-Schema Attender

The utterance-schema attender takes the sequence of utterance embeddings and the set of schema embeddings as input and calculates schema-attended utterance representations and utterance-attended schema representations. In this way, information from the utterances and information from the schemas are fused.

First, the attender constructs an attention matrix, indicating the similarities between utterance embeddings and schema embeddings. Given the i-th utterance token embedding d_i and j-th schema embedding e_j, it calculates the similarity as follows,

$$A(i,j) = r^\mathsf{T}\tanh(W_1 d_i + W_2 e_j), \qquad (1)$$

where r, W_1, W_2 are trainable parameters.

The attender then normalizes each row of matrix A as a probability distribution, to obtain matrix $\overline{A}$. Each row represents the attention weights of schema elements with respect to an utterance token. Then the schema-attended utterance representations are calculated as $D_a = E\overline{A}^\mathsf{T}$. The attender also normalizes each column of matrix A as a probability distribution, to obtain matrix $\widetilde{A}$. Each column represents the attention weights of utterance tokens with respect to a schema element. Then the utterance-attended schema representations are calculated as $E_a = D\widetilde{A}$.

3.4 State Decoder

The state decoder sequentially generates a state representation (semantic frame) for the current turn, which is represented as a sequence of pointers to elements of the schemas and tokens of the utterances (cf., Figure 1). The sequence can then be either re-formalized as a semantic frame in dialogue state tracking[2],

$$[intent; (slot_1, value_1); (slot_2, value_2); ...],$$

or a sequence of labels in NLU (intent-labeling and slot-filling). The pointers point to the elements of intents, slots, and slot values in the schema descriptions (categorical slot values), as well as the tokens in the utterances (non-categorical slot values). The elements in the schemas can be either words or phrases, and the tokens in the utterances form spans for extraction of slot values.

The state decoder is an LSTM using pointer (Vinyals et al., 2015) and attention (Bahdanau et al., 2015). It takes the two representations D_a and E_a as input. At each decode step t, the decoder receives the embedding of the previous item w_{t-1}, the utterance context vector u_t, the schema context vector s_t, and the previous hidden state h_{t-1}, and produces the current hidden state h_t:

$$h_t = \text{LSTM}(w_{t-1}, h_{t-1}, u_t, s_t). \qquad (2)$$

We adopt the attention function in (Bahdanau et al., 2015) to calculate the context vectors as follows,

$$u_t = \text{attend}(h_{t-1}, D_a, D_a), \qquad (3)$$
$$s_t = \text{attend}(h_{t-1}, E_a, E_a). \qquad (4)$$

The decoder then generates a pointer from the set of pointers in the schema elements and the tokens of the utterances on the basis of the hidden state h_t. Specifically, it generates a pointer of item w according to the following distribution,

$$z_w = q^\mathsf{T}\tanh(U_1 h_t + U_2 k_w), \qquad (5)$$
$$P(\#w) = \text{softmax}(z_w), \qquad (6)$$

where $\#w$ is the pointer of item w, k_w is the representation of item w either in the utterance representations D_a or in the schema representations E_a, q, U_1, and U_2 are trainable parameters, and softmax is calculated over all possible pointers.

During decoding, the decoder employs beam search to find the best sequences of pointers in terms of probability of sequence.

3.5 Training

The training of Seq2Seq-DU follows the standard procedure of sequence-to-sequence. The only difference is that it is always conditioned on the schema descriptions. Each instance in training consists of the current utterance and the previous utterances, and the state representation (sequence of pointers) for the current turn. Two pre-trained

---

[2]For simplicity, we assume here that there is only one semantic frame in each turn. In principle, there can be multiple frames.

Characteristics	SGD	MultiWOZ2.2	MultiWOZ2.1	WOZ2.0	DSTC2	M2M	ATIS	SNIPS
No. of domains	16	8	7	1	1	2	-	-
No. of dialogues	16,142	8,438	8438	1,612	600	1,500	4,478	13,084
Total no. of turns	329,964	113,556	113,556	23,354	4,472	14,796	4,478	13,084
Avg. turns per dialogue	20.44	13.46	13.46	14.49	7.45	9.86	1	1
Avg. tokens per turn	9.75	13.13	13.38	8.54	11.24	8.24	11.28	9.09
No. of categorical slots	53	21	37	3	3	0	0	0
No. of non-categorical slots	162	40	0	0	0	14	120	72
Have schema description	Yes	Yes	Yes	No	No	No	No	No
Have unseen schemas in test set	Yes	No	No	No	No	No	No	No

Table 3: Statistics of datasets in experiments. Numbers are those of training datasets.

BERT models are used for representations of utterances and schema descriptions respectively. The BERT models are then fine-tuned in the training process. Cross-entropy loss is utilized to measure the loss of generating a sequence.

4 Experiments

4.1 Datasets

We conduct experiments using the benchmark datasets on task-oriented dialogue. SGD (Rastogi et al., 2019) and MultiWOZ2.2 (Zang et al., 2020) are datasets for DST; they include schemas with categorical slots and non-categorical slots in multiple domains and natural language descriptions on the schemas, as shown in Table 2. In particular, SGD includes unseen schemas in the test set. MultiWOZ2.1 (Eric et al., 2020) is the previous version of MultiWOZ2.2, which only has categorical slots in multiple domains. WOZ2.0 (Wen et al., 2017) and DSTC2 (Henderson et al., 2014) are datasets for DST; they contain schemas with only categorical slots in a single domain. M2M (Shah et al., 2018) is a dataset for DST and it has span annotations for slot values in multiple domains. ATIS (Tur et al., 2010) and SNIPS (Coucke et al., 2018) are datasets for NLU in single-turn dialogues in a single domain. Table 3 gives the statics of datasets in the experiments.

4.2 Baselines and Variants

We make comparison between our approach and the state-of-the-art methods on the datasets.
SGD, MultiWOZ2.2 and MultiWOZ2.1: We compare Seq2SeqDU with six state-of-the-art methods on SGD, MultiWOZ2.2 and MultiWOZ2.1, which utilize schema descriptions, span-based and candidate-based methods, unified seq2seq model and BERT: FastSGT (Noroozi et al., 2020), SGDbaseline (Rastogi et al., 2019), TripPy (Heck et al., 2020), SimpleTOD (Hosseini-Asl et al.,

2020), TRADE (Wu et al., 2019), and DS-DST (Zhang et al., 2019).

WOZ2.0 and DSTC2: Our approach is compared against the state-of-the-art methods on WOZ2.0 and DSTC2, including those using a hierarchical seq2seq model and BERT: COMER (Ren et al., 2019), BERT-DST (Chao and Lane, 2019), StateNet (Ren et al., 2018), GLAD (Zhong et al., 2018), Belief Tracking (Ramadan et al., 2018), and Neural Belief Tracker (Mrkšić et al., 2017).

M2M: We evaluate our approach and the state-of-the-art methods on M2M, which respectively employ a BERT-based architecture and a jointly-trained language understanding model, BERT-DST (Chao and Lane, 2019) and DST+LU (Rastogi et al., 2018).

ATIS and SNIPS: We make comparison between our approach and the state-of-the-art methods on ATIS and SNIPS for NLU within the sequence labeling framework, including Joint BERT (Chen et al., 2019), Slot-Gated (Goo et al., 2018), Atten.-BiRNN (Liu and Lane, 2016), and RNN-LSTM (Hakkani-Tür et al., 2016).

We also include two variants of Seq2Seq-DU. The differences are whether to use the schema descriptions, and the formation of dialogue state.
Seq2Seq-DU-w/oSchema: It is used for datasets that do not have schema descriptions. It only contains utterance encoder and state decoder.
Seq2Seq-DU-SeqLabel: It is used for NLU in a single-turn dialogue. It views the problem as sequence labeling, and only contains the utterance encoder and state decoder.

4.3 Evaluation Measures

We make use of the following metrics in evaluation.
Intent Accuracy: percentage of turns in dialogue for which the intent is correctly identified.
Joint Goal Accuracy: percentage of turns for which all the slots are correctly identified. For non-

categorical slots, a fuzzy matching score is used on SGD and exact match are used on the other datasets to keep the numbers comparable with other works. **Slot F1**: F1 score to evaluate accuracy of slot sequence labeling.

4.4 Training

We use the pre-trained BERT model ([BERT-Base, Uncased]), which has 12 hidden layers of 768 units and 12 self-attention heads to encode utterances and schema descriptions. The hidden size of LSTM decoder is also 768. The dropout probability is 0.1. We also use beam search for decoding, with a beam size of 5. The batch size is set to 8. Adam (Kingma and Ba, 2014) is used for optimization with an initial learning rate of 1e-4. Hyper parameters are chosen using the validation dataset in all cases.

The training curves of all models are shown in Appendix A.

4.5 Experimental Results

Tables 4, 5, 6, and 7 show the results. One can see that Seq2Seq-DU performs significantly better than the baselines in DST and performs equally well as the baselines in NLU.

DST is carried out in different settings in SGD, MultiWOZ2.2, MultiWOZ2.1, WOZ2.0, DSTC2, and M2M. In all cases, Seq2Seq-DU works significantly better than the baselines. The results indicate that Seq2Seq-DU is really a general and effective model for DST, which can be applied to multiple settings. Specifically, Seq2Seq-DU can leverage the schema descriptions for DST when they are available (SGD and MultiWOZ2.2, Multi-WOZ2.1)[3]. It can work well in zero-shot learning to deal with unseen schemas (SGD). It can also effectively handle categorical slots (MultiWOZ2.1, WOZ2.0 and DSTC2) and non-categorical slots (M2M). It appears that the success of Seq2Seq-DU is due to its suitable architecture design with a sequence-to-sequence framework, BERT-based encoders, utterance-schema attender, and pointer generation decoder.

NLU is formalized as sequence labeling in ATIS and SNIPS. Seq2Seq-DU is degenerated to Seq2Seq-DU-SeqLabel, which is equivalent to the baseline of Joint Bert. The results suggest that it is

---

[3]There are better performing systems in the SGD competition. The systems are not based on single methods and thus are not directly comparable with our method.

the case. Specially, the performances of Seq2Seq-DU are comparable with Joint BERT, indicating that Seq2Seq-DU can also be employed in NLU.

Model	SGD		MultiWOZ2.2		MultiWOZ2.1	
	Joint GA	Int Acc	Joint GA	Int Acc	Joint GA	Int Acc
SGD-baseline	0.254	0.906	0.420	-	0.434	-
TRADE	-	-	0.454	-	0.460	-
DS-DST	-	-	0.517	-	0.512	-
FastSGT	0.292	0.903	-	-	-	-
SimpleTOD	-	-	-	-	0.514	-
TripPy	-	-	0.535	-	0.553	-
Seq2Seq-DU	**0.301**	**0.910**	**0.544**	**0.909**	**0.561**	**0.911**

Table 4: Accuracies of Seq2Seq-DU and baselines on SGD, MultiWOZ2.2 and MultiWOZ2.1 datasets. Seq2Seq-DU outperforms baselines in terms of all metrics.

Model	WOZ2.0	DSTC2
	Joint GA	Joint GA
Neural Belief Tracker	0.842	0.734
Belief Tracking	0.855	-
GLAD	0.881	0.745
StateNet	0.889	0.755
BERT-DST	0.877	0.693
COMER	0.886	-
Seq2Seq-DU-w/oSchema	**0.912**	**0.850**

Table 5: Accuracies of Seq2Seq-DU and baselines on WOZ2.0 and DSTC2 datasets. Seq2Seq-DU-w/oSchema performs significantly better than the baselines.

Model	M2M	
	Joint GA	Int Acc
DST+LU	0.767	-
BERT-DST	0.869	-
Seq2Seq-DU-w/oSchema	**0.909**	**0.997**

Table 6: Accuracies of Seq2Seq-DU and baselines on M2M dataset. Seq2Seq-DU-w/oSchema significantly outperforms the baselines.

Model	ATIS		SNIPS	
	Slot F1	Int Acc	Slot F1	Int Acc
RNN-LSTM	0.943	0.926	0.873	0.969
Atten.-BiRNN	0.942	0.911	0.878	0.967
Slot-Gated	0.952	0.941	0.888	0.970
Joint BERT	**0.961**	**0.975**	**0.970**	0.986
Seq2Seq-DU-SeqLabel	0.955	0.968	0.965	**0.990**

Table 7: Accuracies of Seq2Seq-DU and baselines on ATIS and SNIPS datasets. Seq2Seq-DU-SeqLabel performs comparably with Joint BERT.

4.6 Ablation Study

We also conduct ablation study on Seq2Seq-DU. We validate the effects of three factors: BERT-based encoder, utterance-schema attention, and pointer generation decoder. The results indicate that all the components of Seq2Seq-DU are indispensable.

Effect of BERT

To investigate the effectiveness of using BERT in the utterance encoder and schema encoder, we replace BERT with Bi-directional LSTM and run the model on SGD and MultiWOZ2.2. As shown in Figure 3, the performance of the BiLSTM-based model Seq2Seq-DU-w/oBert in terms of Joint GA and Int. Acc decreases significantly compared with Seq2Seq-DU. It indicates that the BERT-based encoders can create and utilize more accurate representations for dialogue understanding.

Effect of Attention

To investigate the effectiveness of using attention, we compare Seq2Seq-DU with Seq2Seq-DU-w/oAttention which eliminates the attention mechanism, Seq2Seq-DU-w/SchemaAtt which only contains the utterance-attended schema representations, and Seq2Seq-DU-w/UtteranceAtt which only contains the schema-attended utterance representations. Figure 3 shows the results on SGD and MultiWOZ2.2 in terms of Joint GA and Int. Acc. One can observe that without attention the performances deteriorate considerably. In addition, the performances of unidirectional attentions are inferior to the performance of bidirectional attention. Thus, utilization of bidirectional attention between utterances and schema descriptions is desriable.

Effect of Pointer Generation

To investigate the effectiveness of the pointer generation mechanism, we directly generate words from the vocabulary instead of generating pointers in the decoding process. Figure 3 also shows the results of Seq2Seq-DU-w/oPointer on SGD and MultiWOZ2.2 in terms of Joint GA and Int. Acc. From the results we can see that pointer generation is crucial for coping with unseen schemas. In SGD which contains a large number of unseen schemas in the test set, there is significant performance degradation without pointer generation. The results on MultiWOZ2.2, which does not have unseen schemas in the test set, show pointer generation can also make significant improvement on
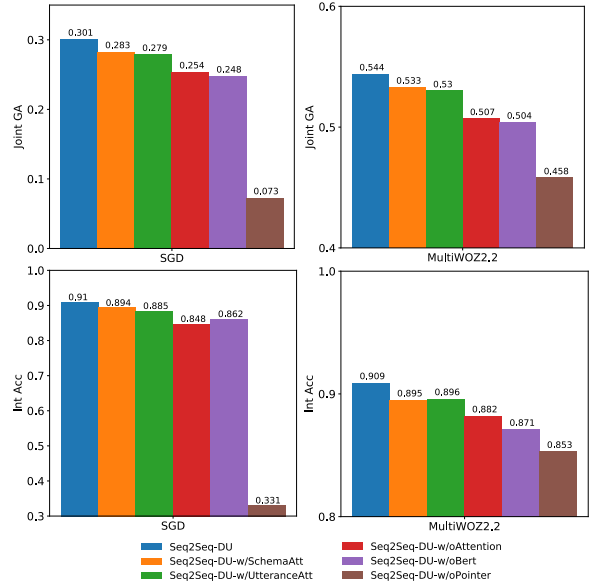

Figure 3: Ablation study results of Seq2Seq-DU with respect to BERT, attention, and pointer generation on SGD and MultiWOZ2.2.

already seen schemas by making full use of schema descriptions.

4.7 Discussions

Case Study

We make qualitative analysis on the results of Seq2Seq-DU and SGD-baseline on SGD and MultiWOZ2.2. We find that Seq2Seq-DU can make more accurate inference of dialogue states by leveraging the relations existing in the utterances and schema descriptions. For example, in the first case in Table 8, the user wants to find a cheap guesthouse. Seq2Seq-DU can correctly infer that the hotel type is "guesthouse" by referring to the relation between "hotel-pricerange" and "hotel-type". In the second case, the user wants to rent a room with in-unit laundry. In the dataset, a user who intends to rent a room will care more about the laundry property. Seq2Seq-DU can effectively extract the relation between "intent" and "in-unit-laundry", yielding a correct result. In contrast, SGD-baseline does not model the relations in the schemas, and thus it cannot properly infer the values of "hotel-type" and "in-unit-laundry".

Dealing with Unseen Schemas

We analyze the zero-shot learning ability of Seq2Seq-DU. Table 9 presents the accuracies of Seq2Seq-DU in different domains on SGD. (Note that only SGD has unseen schemas in test set.) We observe that the best performances can be obtained

ID	Dialogue Utterance	Dialogue State	State Predictions of SGD-baseline	State Predictions of Seq2Seq-DU
1	*User*: I wanna rent a place in Campbell. *Sys*: How many baths? *User*: One bath is fine. *Sys*: How many bedrooms? *User*: One bedroom is fine. It also needs in-unit laundry.	"area": Campbell; "in-unit-laundry": True; "intent": rent; "number-of-baths": 1; "number-of-beds": 1; "active-intent": FindHomeByArea;	"area": Campbell; "in-unit-laundry": – ; "intent": rent; "number-of-baths": 1; "number-of-beds": 1; "active-intent": FindHomeByArea;	"area": Campbell; "in-unit-laundry": True; "intent": rent; "number-of-baths": 1; "number-of-beds": 1; "active-intent": FindHomeByArea;
2	*User*: The location isn't really important. It does need to be cheap though, and preferably a guesthouse.	"hotel-area": dontcare ; "hotel-pricerange": cheap; " hotel-type": guesthouse; "active intent": find-hotel;	"hotel-area": dontcare ; "hotel-pricerange": cheap; "hotel-type": hotel; "active intent": find-hotel;	"hotel-area": dontcare; "hotel-pricerange": cheap; "hotel-type": guesthouse; "active intent": find-hotel;

Table 8: Case study on Seq2Seq-DU and SGD-baseline on SGD and MultiWOZ2.2. The first example is from SGD and the second is from MultiWOZ2.2. The underlined slot-value pairs represent the ground truth states. The slot-value pairs in blue are correctly predicted ones, while the slot-value pairs in red are incorrectly predicted ones.

in the domains with all seen schemas. The domains that have more partially seen schemas achieve higher accuracies, such as "Hotels", "Movies", "Services". The accuracies decline in the domains with more unseen schemas, such as "Messaging" and "RentalCars". We conclude that Seq2Seq-DU can perform zero-shot learning across domains. However, the ability still needs enhancement.

Domain	Joint GA	Int Acc	Domain	Joint GA	Int Acc
*Messaging**	0.0489	0.3510	*Media**	0.2307	0.9065
*RentalCars**	0.0625	0.7901	*Events**	0.3186	0.9327
*Payment**	0.0719	0.5835	*Hotels***	0.3396	0.9891
*Music**	0.1234	0.9438	*Movies***	0.4386	0.7836
*Restaurants**	0.1295	0.9627	*Travel*	0.4490	0.9966
*Flights**	0.1589	0.9649	*Services***	0.4774	0.9842
*Trains**	0.1683	0.9257	*Alarm**	0.5567	0.5768
*Buses**	0.1684	0.8805	*Weather*	0.5792	0.9965
Homes	0.2275	0.9081	*RideSharing*	0.6702	0.9991

Table 9: Accuracy of Seq2Seq-DU in each domain on SGD test set. Domains marked with '*' are those for which the schemas in the test set are not present in the training set. Domains marked with '**' have both the unseen and seen schemas. For other domains, the schemas in the test set are also seen in the training set.

Categorical Slots and Non-categorical Slots

Table 10 shows the accuracies of Seq2Seq-DU and the baselines with respect to categorical and non-categorical slots on SGD and MultiWOZ2.2. (We did not compare with FastSGT on SGD dataset due to unavailability of the codes.) One can see that Seq2Seq-DU can effectively deal with both categorical and non-categorical slots. Furthermore, Seq2Seq-DU demonstrates higher accuracies on categorical slots than non-categorical slots. We conjecture that it is due to the co-occurrences of categorical slot values in both the dialogue history and the schema descriptions. The utterance-schema

attention can more easily capture the relations between the values.

Model	SGD		MultiWOZ2.2	
	Categorical-Joint-GA	Noncategorical-Joint-GA	Categorical-Joint-GA	Noncategorical-Joint-GA
TRADE	-	-	0.628	0.666
SGD-baseline	0.513	0.361	0.570	0.661
DS-DST	-	-	0.706	0.701
FastSGT	not available	not available	-	-
TripPy	-	-	0.684	**0.733**
Seq2Seq-DU	**0.578**	**0.393**	**0.758**	0.711

Table 10: Accuracies of Seq2Seq-DU and baselines with respect to categorical and non-categorical slots on SGD and MultiWOZ2.2.

5 Conclusion

We have proposed a new approach to dialogue state tracking. The approach, referred to as Seq2Seq-DU, takes dialogue state tracking (DST) as a problem of transforming all the utterances in a dialogue into semantic frames (state representations) on the basis of schema descriptions. Seq2Seq-DU is unique in that within the sequence to sequence framework it employs BERT in encoding of utterances and schema descriptions respectively and generates pointers in decoding of dialogue state. Seq2Seq-DU is a global, reprentable, and scalable model for DST as well as NLU (natural language understanding). Experimental results show that Seq2Seq-DU significantly outperforms the state-of-the-arts methods in DST on the benchmark datasets of SGD, MultiWOZ2.2, MultiWOZ2.1, WOZ2.0, DSTC2, M2M, and performs as well as the state-of-the-arts in NLU on the benchmark datasets of ATIS and SNIPS.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In ICLR.

Guan-Lin Chao and Ian Lane. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. In INTERSPEECH.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. Acm Sigkdd Explorations Newsletter.

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

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. arXiv preprint arXiv:1902.10909.

Zhi Chen, Lu Chen, Zihan Xu, Yanbin Zhao, Su Zhu, and Kai Yu. 2020b. Credit: Coarse-to-fine sequence generation for dialogue state tracking. arXiv preprint arXiv:2009.10435.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. In CoRR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In LREC.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In NAACL.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In INTERSPEECH.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gašić. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In ACL.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In SIGDIAL.

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.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. ACM Transactions on Information Systems (TOIS).

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In CoRR.

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 ACL.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In INTERSPEECH.

Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In ACL.

Vahid Noroozi, Yang Zhang, Evelina Bakhturina, and Tomasz Kornuta. 2020. A fast and robust bert-based dialogue state tracker for schema-guided dialogue dataset. arXiv preprint arXiv:2008.12335.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In ICLR.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In ACL.

Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. 2018. Multi-task learning for joint language understanding and dialogue state tracking. In SIGDIAL.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In ASRU.

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 EMNLP.

Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. Towards universal dialogue state tracking. In EMNLP.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. arXiv preprint arXiv:1801.04871.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In SLT.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In NIPS.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, 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 EACL.

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 ACL.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In ACL.

Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. 2014. Recurrent conditional random field for language understanding. In ICASSP.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In ACL.

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.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, Minlie Huang, and Xiaoyan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. Science China Technological Sciences.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In ACL.

A Training Curves

Figure 4 shows the training losses of Seq2Seq-DU on the training datasets, while Figure 5 shows the accuracies of Seq2Seq-DU on the test sets during training. We regard training convergence when the fluctuation of loss is less than 0.01 for consecutive 20 thousand steps. Seq2Seq-DU converges at the 180k-th step on SGD, MultiWOZ2.2, and Multi-WOZ2.1. Seq2Seq-DU-w/oSchema converges at the 150k-th step on WOZ2.0 and at the 140k-th step on DSTC2, and M2M. Furthermore, Seq2Seq-DU-SeqLabel converges at the 130k-th step on ATIS and SNIPS. These are consistent with the general trends in machine learning that more complex models are more difficult to train.
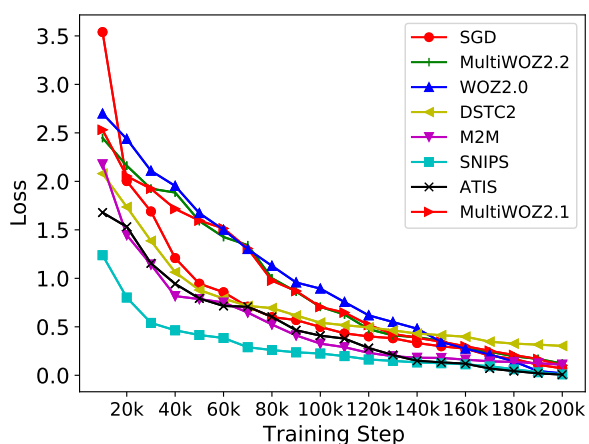

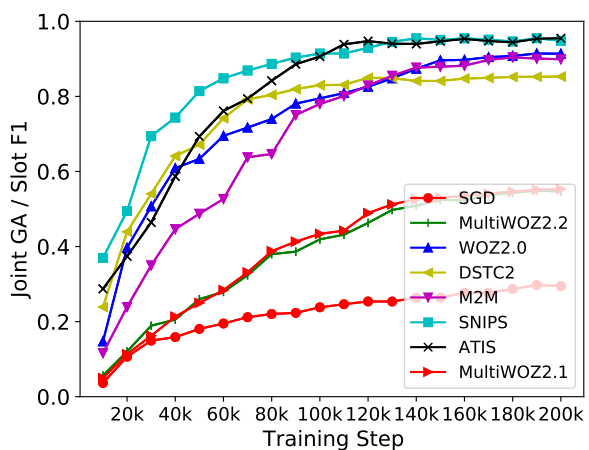Figure 4: Training losses of Seq2Seq-DU on all training datasets.

Figure 5: Accuracies of Seq2Seq-DU in terms of Join GA / Slot F1 on all test sets.