Dialogue State Tracking with Text Infilling and Curriculum Training
NLP 244 Final Project Report

Brian Mak
bmak2@ucsc.edu

Diji Yang
dyang39@ucsc.edu

Ken Ting
cting3@ucsc.edu

Abstract

Dialog state tracking task can be really important for either generating responses and performing action for users. This paper has conducted experiments with the usage of curriculum training on text infilling tasks with different difficulties for dialogue state tracking.

1 Task Definition

Dialog state tracking is one of the component of task-oriented dialog system, which is responsible for estimating the user's goal at each time step. It will take dialogues between the user and the conversational agent as the input and predict the state of the corresponding dialog as the output. The input dialog U_t will contain all previous context until the time step t, where $U_t = \{u_0, u_1, u_2, ..., u_t\}$ and u_t denotes the utterance sentence at time step t. The dialog state indicates the current status that the agent understand about what the user requires. For example, if the user is having dialogues of asking the agent to book the hotel, the agent will store the state of domain of "hotel" and slots related to the "hotel", such as the hotel's name or the hotel's location. The dialog state at the time t will consider the entire context until t in the dialog. As mentioned in Zhang et al., 2020, most recent works considered belief state for dialog state representation, which is composed of slot-value pairs. And the problem can be formulated as a multi-task classification task:

$$p_i(d_{i,t}|u_1, u_2, ..., u_t) \tag{1}$$

where for each slot i, its corresponding tracker p_i will predict the value $d_{i,t}$ of the slot in the state given previous utterances $u_1, u_2, ..., u_t$.

Terminology In this report, we use DS to indicate the Dialog State, NLG for Natural Language Generation task, and XLM refers to TML (Translation Language Modeling) from Lample and Conneau (2019).

2 Previous Approach

2.1 TripPy

Heck et al. (2020) has proposed a triple copy mechanism on the utterance to the slot for dialogue state tracking. The mechanism considered three cases: span detection, copying system informed memory, and referring to previous dialogue states. These mechanism takes advantage on the span slot filling and also consider about the information in the history or past states. The mechanism has helped some further researches reach state-of-the-art result.

2.2 SimpleTOD

Hosseini-Asl et al. (2020) has proposed the method, SimpleTOD, which formulate the task-oriented dialogue system into a simple sequence-to-sequence task. It trained GPT-2 with Causal Language Modeling objective on all the sub-tasks of task-oriented dialogues, including NLG, DST, and action generation. For inference, it will generate the predicted output token by token and stop at the end of a subtask, such as end token of belief state for DST and then start generating again after getting the results from DB query. We will take the idea of the training schema, especially for DST and NLG part, for training our proposed model.

3 Motivation for this Approach

Motivated by SimpleTOD, it is reasonable to think dialog state is a sentence that comes after an utterance. However, instead of make the whole dialog turn as a language model (next token/sentence prediction), we would like to make DST task as a machine translation task. To be more explicit, we define the Dialogue (utterances) as the Language 1, the Dialog State (domain-slot-value) as language 2, and the task as the translation between these two languages. We form DST task in a supervised way and use those two languages as the parallel data. The reason behind this idea is part from the nature of the languages, i.e., intuitively, the dialog state is not necessarily depends on the utterances level representation. Some importance word/span, in most cases, is enough to assign a high probability score to the correct prediction. In another word, some word in the sentence may totally useless for the dialog state prediction. While any neural model with attention mechanism allows model to capture these things, XLM as mentioned in 4.1 demonstrates that it is particularly good at learn this ”translation dictionary” in cross-lingual scenarios. Model can learn a stronger correspondence between dialogues and the dialog states. Moreover, under this setting, we also expect our model can do the translation in both way, that is, from language 1 to 2 and also 2 to 1.

4 Related Works

Besides SimpleTOD, our design relies on XLM and BART (Lewis et al., 2019).

4.1 XLM

In our idea of DST as MT, Lample and Conneau (2019) inspire us a lot, especially TLM. TLM is an extension of mask language modeling, but it works on concatenated parallel sentences. The position embedding of the language 2 are reset to record the connection point of two sentences. The tokens from both languages can be masked. To predict a masked word, the model can either attend to surrounding words in the source language or the translation. This design encourages model to align the representation between source and target language.

4.2 BART

Our proposed method will try to fine-tune BART(Lewis et al., 2019), which is a pretrained denoised sequence-to-sequence model. It can be considered as the architecture of using BERT as the encoder and GPT as the decoder. It has been pretrained by several noising objectives which is try to corrupt sentence in different ways. BART will try to reconstruct the text and learn how to denoise the corrupted text and guessing the original form. Among all the pretraining objective, it performed the best on the downstream task with text infilling pretraining, which we will take this property for constructing our learning objective on masking dialogue states for DST and generated texts for NLG.

5 Dataset

MultiWOZ(Budzianowski et al., 2020) dataset is a large-scale human-human conversational corpus. It contains 8,438 conversational dialogues with multiple turns. The dataset is able to evaluate models of three type of tasks, Dialogue State Tracking, Dialogue-Context-to-Text Generation, and Dialogue-Act-to-Text Generation. In this work, we will focus on the evaluation of Dialogue State Tracking.

5.1 Statistics

The dataset spans over 7 domains, Attraction, Hospital, Police, Hotel, Restaurant, Taxi, and Train. 3,406 of dialogues are labeled with single domain(1-2 domains) and 7,032 are with multiple domains(2-5 domains). There are other detailed stats in the figure 1. See example in figure 2.

5.2 MultiWOZ 2.1

We will perform the training and evaluation on MultiWoz 2.1(Eric et al., 2019). Since the dataset is a large corpus, it contains different kind of noises that could cause negative impact to the performance. The 2.1 version of MultiWOZ(Eric et al., 2019) have fixed several annotation noises and apply the augmented data that provided by a follow-up work(Lee et al., 2019). Many of state-of-arts in Dialogue State Tracking task are using this corpus for their performance evaluation.

6 Design

Figure 3 shows the structure for the entire working pipeline. In this section we split the detailed design into three parts: model selection, finetuning method and curriculum training scheme.

6.1 Model Selection

The model architecture we used for this work is BART. Since we only want to take advantage of the text infilling capabilities of BART, we need not make any modifications to the architecture itself. Below we will describe how the system should behave at inference time and then how the system will be trained. At inference time, our system will use text infilling to predict the dialog state of the conversation. This approach is almost identical to the approach used by (Hosseini-Asl et al., 2020) with

Metric	DSTC2	SFX	WOZ2.0	FRAMES	KVRET	M2M	MultiWOZ
# Dialogues	1,612	1,006	600	1,369	2,425	1,500	**8,438**
Total # turns	23,354	12,396	4,472	19,986	12,732	14,796	**113, 556**
Total # tokens	199,431	108,975	50,264	251,867	102,077	121,977	**1,490,615**
Avg. turns per dialogue	14.49	12.32	7.45	**14.60**	5.25	9.86	13.46
Avg. tokens per turn	8.54	8.79	11.24	12.60	8.02	8.24	**13.13**
Total unique tokens	986	1,473	2,142	12,043	2,842	1,008	**23689**
# Slots	8	14	4	**61**	13	14	24
# Values	212	1847	99	3871	1363	138	**4510**

Figure 1: Comparison of our corpus to similar data sets. Numbers in bold indicate best value for the respective metric. The numbers are provided for the training part of data except for FRAMES data-set were such division was not defined. (Budzianowski et al., 2020)

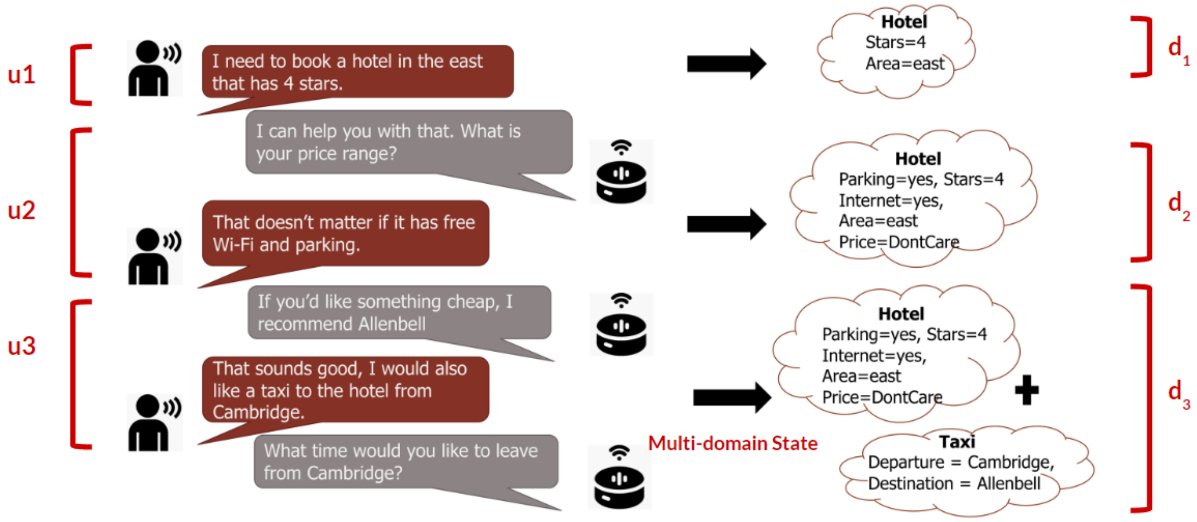
u1
I need to book a hotel in the east that has 4 stars.
Hotel
Stars=4
Area=east
d1
I can help you with that. What is your price range?
u2
That doesn't matter if it has free Wi-Fi and parking.
Hotel
Parking=yes, Stars=4
Internet=yes,
Area=east
Price=DontCare
d2
If you'd like something cheap, I recommend Allenbell
u3
That sounds good, I would also like a taxi to the hotel from Cambridge.
What time would you like to leave from Cambridge?
Multi-domain State
Hotel
Parking=yes, Stars=4
Internet=yes,
Area=east
Price=DontCare
Taxi
Departure = Cambridge,
Destination = Allenbell
d3

Figure 2: Example of dialog/state pair of MultiWOz dataset.

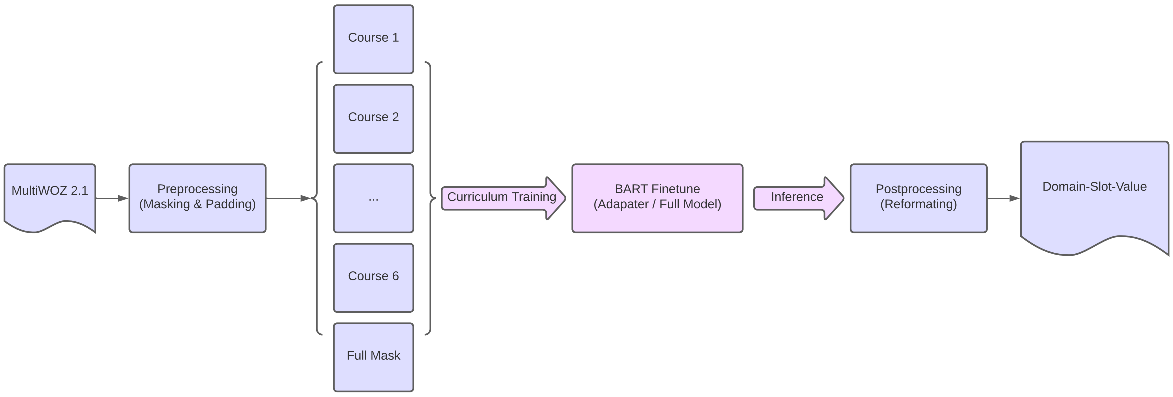
MultiWOZ 2.1
Preprocessing (Masking & Padding)
Course 1
Course 2
...
Course 6
Full Mask
Curriculum Training
BART Finetune (Adapater / Full Model)
Inference
Postprocessing (Reformating)
Domain-Slot-Value

Figure 3: Structure for the entire working pipeline

a few modifications. The first modification only pertains to the dialog state tracking task. (Hosseini-Asl et al., 2020) used causal language modelling to predict the next dialog state in a serialized format (Figure 4). This includes generating both domain slot name and corresponding value of every slot value in the belief state.

We instead used text infilling as a means of predicting the dialog state. This mean that we provided the model with the names of all belief slot names

Belief State	[belief] *domain slot_name value, domain slot_name value, . . .* [endofbelief]

Figure 4: SimpleTOD belief state format
z

and mask the values. The model then infill the mask values, producing the complete dialog state.

6.2 Model Fine-tuning

Follow the pretrain finetune paradigm, based on the MultiWOZ dataset, full model finetuning can be applied to BART model to pursuit better performance on DST task. Motivated by parameter efficiency method, instead of full model finetuning, the extra adapter layers can be trained and added into originally BART model while other parts of model are all fixed.

6.3 Training scheme

The novelty of our design comes from the training scheme. We use a combination of ideas from (Bengio et al.) and (Lample and Conneau, 2019). We will use the TLM training objective from (Lample and Conneau, 2019). To be specific, for the task of dialog state tracking we will feed dialog contexts and dialog states to our model with some token spans masked out. Token spans from either the dialog context or the dialog state will be masked, but not both at at the same time. We then will train the model to predict the masked token span. Similarly to how the XLM model learns to use information from the sentence in the unmasked language to predict masked tokens in the masked language, we expect our model to use the information present in the utterance to predict dialog state and vice versa. The same training objective framework can be used for natural language generation. Here, we will feed in parallel intended actions and system responses with token spans masked. The same concepts from the dialog state tracking task apply here except the goal is to generate the utterance instead of the dialog state. We plan to use the same model for both tasks as we expect the translation between dialog state and user utterance to be similar to the translation between intended action and system utterance (the same applies for the inverse translations).

We also plan to use a curriculum learning approach layered on top of the TLM modeling objective. This means that we will start with masking very few tokens, perhaps just one entity. Once the system performs well on the ”easy” task, we will gradually increase the difficulty of the training objective by increasing the number of tokens masked. An idea is to mask multiple entities after a single one, then text spans that cover a large portion of the utterance/dialog state/intended action, then finally the entire utterance/dialog state/intended action. The exact nature of our curriculum will have to be experimented with. We will also experiment with our criterion of what it means for the system to perform well enough on a task before we increase the level. One initial idea is to set a threshold on validation accuracy before moving the the next level of difficulty. This is another hyperparameter to be tuned. By the end of this training we should have a model that can predict full dialog states and system responses.

7 Evaluation

For evaluating the performance of dialog state tracking, we will use two automatic evaluation metrics which have been widely used in this task: the joint goal accuracy and the slot accuracy(Takanobu et al., 2020). The joint goal accuracy will check whether the state of each turn is exactly matching the ground truth state for all the slots. The output is considered correct if and only if all the predicted values are exactly matching the correct answer in one single turn. On the other hand, the slot accuracy will individually compare the slot value to the ground truth value. In the multi-domain dialogue state tracking, it will compare the slot accuracy of the (domain, slot, value) triplet.
However, the slot accuracy may give results that is too optimistic and joint goal accuracy can be too strict for the evaluation. Kim et al. (2022a) has proposed a metric, relative slot accuracy, to complement these drawbacks. The relative slot accuracy does not depend on predefined slots and only calculate the scores by considering the slots being affected by the current state. It is basically same as slot accuracy but only considering the slots appeared in both predicted and gold states.

8 Experiment

As shown in table 1, we trained serval different models using adapter or full model finetuning. The training Strategy indicates weather with or without curriculum training. With same Nvidia 3090, adapter can be updated in around 20 minutes for single epoch, yet full model fine-tuning takes about 40 minutes per epoch. We also evaluated BART model without any task specific finetuning to report the zero-shot result on DST.

8.1 Masking scheme

For the curriculum training, we have defined different masking schemes as different difficulties of the course:

Masking Delta This scheme will only mask the dialogue state value that only informed in the utterance of the current step, which also indicate the difference of the current state and the previous state.

Masking Belief This scheme will mask the current dialogue state value randomly with a given probability. The probability represent how many proportion of the full set of dialogue values should be masked.

Masking Previous Belief This scheme will mask the previous dialogue state values randomly with a given probability. The probability represent how many proportion of the full set of dialogue values should be masked.

Masking Both Belief This scheme will mask both the current and previous dialogue state values randomly with a given probability. The probability represent how many proportion of the full set of dialogue values should be masked.

Masking Belief Context This scheme will only mask the utterance tokens that is associated with the current belief value.

Masking Utterance Same as the masking belief, but this scheme is focusing on utterance. It will randomly pick a proportion of the utterance tokens depending on given probability and mask them.

The experiment will start the curriculum training with 1 epoch for each course. Afterward, they will be fine-tuned on the masking both belief scheme with belief values are fully masked(100%) until converged. For the beginning curriculum, we have designed:

Curriculum 1 Initially, we decided to train for both dialogue state and response generation tasks. So we have design a series of courses that may include the understanding of utterance generation. The curriculum will be:

- Masking Delta
- Masking Belief(15%)
- Masking Utterance(15%)
- Masking Belief Context
- Masking Belief(50%)
- Masking Utterance(50%)

Curriculum 2 However, we only perform the evaluation on the dialogue state tracking in the end, so we modified the curriculum into dialogue state tracking focused objective. It became:

- Masking Delta
- Masking Belief(25%)
- Masking Previous Belief(25%)
- Masking Belief(50%)
- Masking Previous Belief(50%)
- Masking Both Belief(50%)

9 Result

9.1 Loss Analysis

The figure 5 shows the training loss and validation loss of the BART adapter with and without the curriculum 1 training. The loss curves for training yield some interesting trends. We can see that when the curriculum is easy (25% masked or deltas masked), the loss converges much more quickly than the no curriculum training. This is to be expected because it is an easier task. As the tasks become more difficult, the training loss starts to jump up (epochs 3 and 5). In the final stage where all fields are masked, the curriculum loss exceeds the no curriculum loss. It then converges to around the same level. The validation loss shows somewhat of a different story, with the curriculum loss maintaining a lower value than the no curriculum loss when going to the harder tasks. This along with our curriculum training's superior results show evidence for the fact that the curriculum trained network did find a better local minimum than the non curriculum trained network. Thus we have some evidence to suggest that the curriculum training worked in aligning with the theory.

However, since we only want to focus on dialogue state tracking only, we also track the loss of curriculum 2 training. The figure 6 shows the training loss and validation loss of the BART full model with and without the curriculum 2 training.

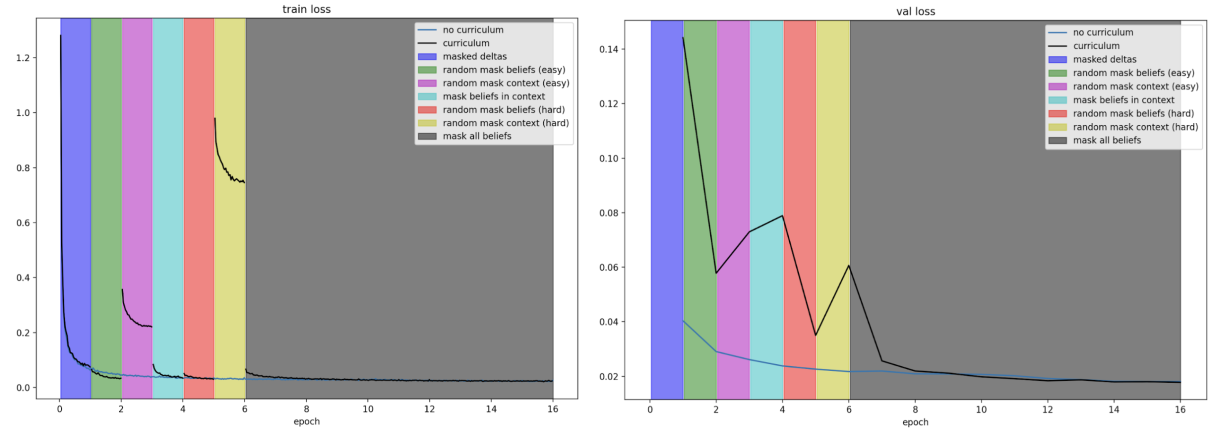

Figure 5: BART Adapter with/without Curriculum 1 Loss

The training loss of curriculum training does reflect different difficulty of the courses, which may having higher gap of the training loss if the difficulty is changing too much. For the validation loss, since the model without curriculum is converged and early stopped, we can not compare the losses for the further epoch. But the interesting thing is, the model with curriculum training has converged even before the final stage fine-tuning, which is full masking. It produced a slightly better validation loss than the model without curriculum, which may indicate that the curriculum training does give the model a better gradient for reaching the optimal point.

9.2 Performance Comparison

The performance comparison results can be found in the table 1. While our model did not achieve the same results as SimpleTOD and Tripy-CoCo, we can still draw some positive results from our experiments. We can see that our text infilling method did in fact work, scoring a very reasonable 47.75% joint accuracy when finetuning BART. If we had more computing resources and time we may have been able to raise this score by doing some hyperparameter tuning on the training parameters. This score, however, is within the ballpark of SimpleTOD and TripPy-Coco.

The results of our curriculum training shows that this strategy increased the accuracy of our models in all metrics. In particular, it raised the joint accuracy of the models by a relatively large margin (compared to the other metrics). Furthermore, we suspect that with hyperparameter tuning of training parameters and curriculum courses, along with training each course to convergence, we could get a more competitive score with the other SOTA models. We already see that our slot accuracy on the fine tuned models beat out SimpleTOD and TripPy-CoCo.

9.3 Error Analysis

When we take a closer look at how to curriculum and no curriculum models perform by slot value, we see that the results are very competitive for most slots(Figure 7). In general, the curriculum trained model seems to slightly outperform the non curriculum trained model in all fields. In the fields ”train departure” and ”train book people”, the curriculum trained model performers much better. Upon further analysis, the non curriculum model seems to predict ”not mentioned” for these fields while the curriculum trained model predicts correctly (in the cases were the curriculum model did better). We are still unsure about the reasons why this is the case but it shows that curriculum training can be particularly suited for some particular slots.

9.4 Turns Analysis

For the performance of each turn of the models that are with and without curriculum training, we found an interesting trending according to the figure 8. The performance of the model without curriculum training have better score on the smaller turns for all the metrics. However, when the turns number increase to the intermediate number of the turns, the model with curriculum training has performed better than the model without curriculum training. It is observed that the curriculum training help model better addressed the difficult tasks, however, the model without curriculum training has learned the task more straightforwardly so that

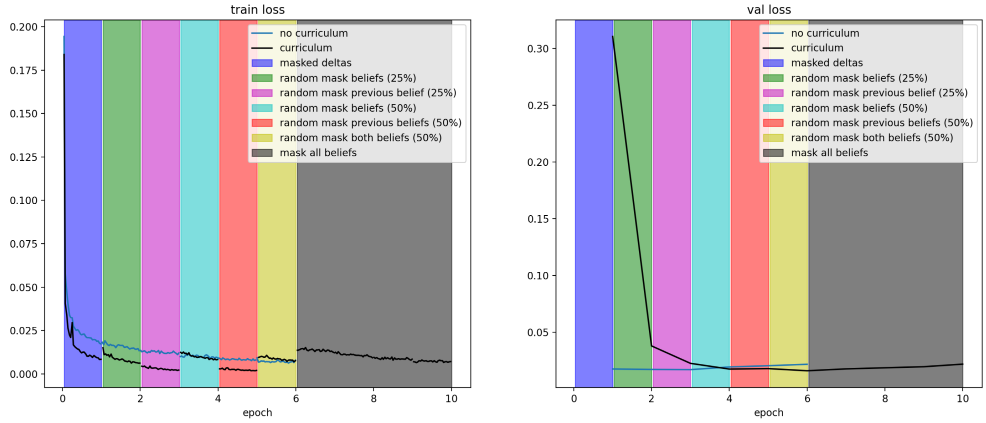

Figure 6: BART Full Model with/without Curriculum 2 Loss

Model	Training Strategy	Training Epoch	Joint Acc	Slot Acc	Relative Slot Acc
BART	Zero-shot		1.44	85.15	0
BART-adapter	None	10	37.25	96.40	77.46
BART-adapter	Curriculum Training	16	42.72	96.9	80.45
BART-finetune	None	16	47.75	97.47	83.89
BART-finetune	Curriculum Training	16	48.02	97.47	84
SimpleTOD	Reproduced by Kim et al. (2022b)		56.05	97.61	87.97
TripPy-CoCo	Reproduced by Kim et al. (2022b)		61.31	97.07	84.32

Table 1: Model performance on MultiWOZ 2.1 with various evaluation metrics. Our training is done by single Nvidia 3090. TripPy-CoCo (Li et al., 2020) is the current state-of-the-art for MultiWOZ 2.1 using Joint Accuracy measurement.

it performed slightly better on the easiest tasks. So the curriculum design may have effect on tasks performance of different difficulty. It would be an interesting study to find out what kind of design of curriculum will have what specific influence on the performance.

10 Conclusion

In this final report, we review the task definition, the entire project flow and report our experimental results. Based on the observed performance, we analyze the impact of curriculum training and compare the advantages and drawbacks of the adapter and full model finetuning method. In MultiWOZ 2.1 dataset, our best model achieves a comparative result to the current state-of-the-art model in terms of slot accuracy, but with much less training time. Empirically, our result can be further improved by simply refining the masking scheme and design a more efficient curriculum.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2020. Multiwoz – a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines.

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.

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

Takyoung Kim, Hoonsang Yoon, Yukyung Lee, Pilsung Kang, and Misuk Kim. 2022a. Mismatch between multi-turn dialogue and its evaluation metric in dialogue state tracking. *arXiv preprint arXiv:2203.03123*.

Takyoung Kim, Hoonsang Yoon, Yukyung Lee, Pilsung Kang, and Misuk Kim. 2022b. Mismatch between multi-turn dialogue and its evaluation metric in dialogue state tracking. *arXiv preprint arXiv:2203.03123*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Xiang Li, Yaoqin Zhang, Zheng Zhang, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019. Convlab: Multi-domain end-to-end dialog system platform.

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.

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

Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao, and Minlie Huang. 2020. Is your goal-oriented dialog model performing really well? empirical analysis of system-wise evaluation. *CoRR*, abs/2005.07362.

Zheng Zhang, Ryuichi Takanobu, Minlie Huang, and Xiaoyan Zhu. 2020. Recent advances and challenges in task-oriented dialog system. *CoRR*, abs/2003.07490.

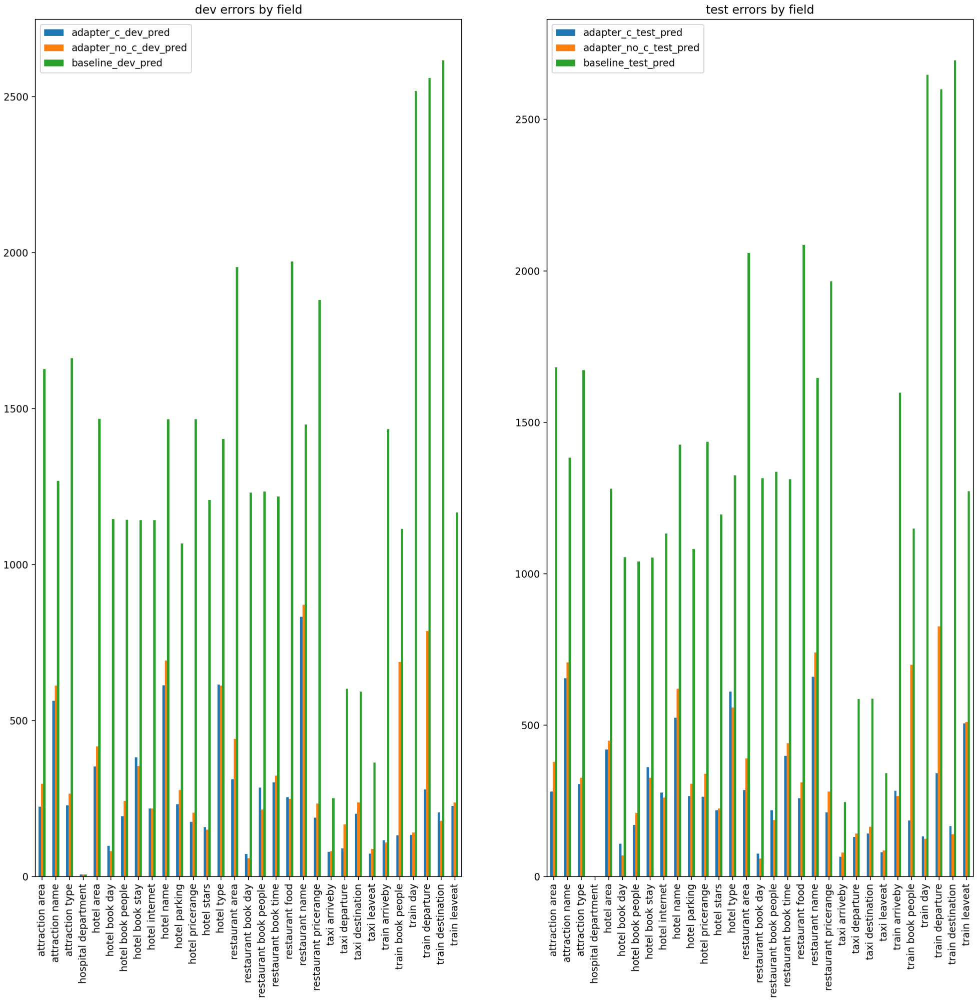

Figure 7: BART finetuned error analysis by field

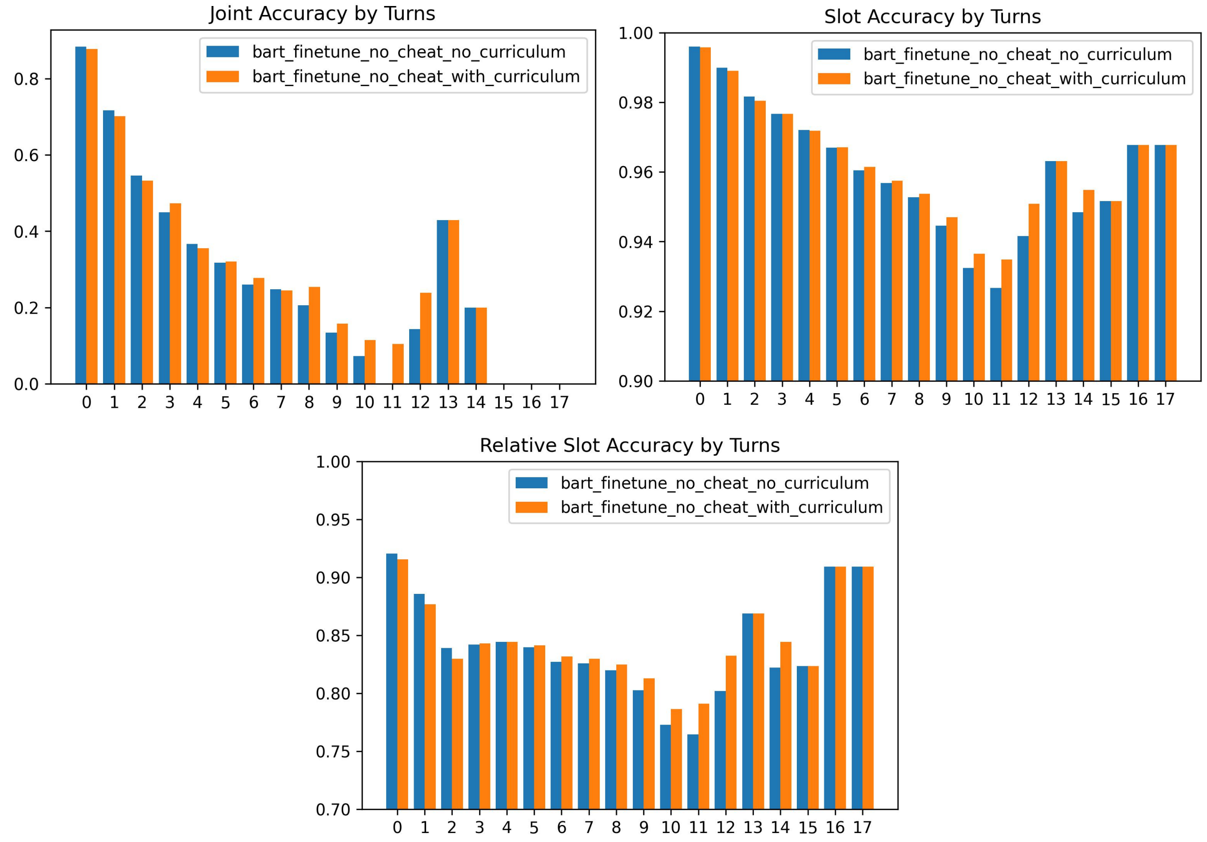

Figure 8: Accuracy By Turns, Model: BART Full Model with/without curriculum training, x-axis represent number of turns included in the input dialogue