Using Masked Span Language Modeling for Multi-Domain Dialogue State Tracking

Xiaolong Xu[1], **Jin Li**[2], **Guo Chen**[3], **Gaoya Jin**[4]
[1] Research Institute of Tsinghua University in Shenzhen
[2] Botomatic Inc.
[3] Carnegie Mellon University
[4] Lanzhou Jiaotong University
xiaolong06.xu@gmail.com, jin.li@btmatic.com, gchen2@cs.cmu.edu, 0219646@stu.lzjtu.edu.cn

Abstract

This paper describes our approach in the Tenth Dialogue System Technology Challenge (DSTC10) Track 2, specifically in multi-domain dialogue state tracking (DST) on spoken conversations. Although attention-based pre-trained language models are beneficial to the task of DST, pre-trained masked language models create a misalignment between pre-training and DST tasks, since the [MASK] token does not appear during fine-tuning. Consequently, we introduced masked span language modeling training objectives based on T5 for DST, and we further adopted multi-task pre-training and post-processing to compensate for the limited automated speech recognition (ASR) data. Our approach is the best performing single model and is ranked second in the objective evaluation of DSTC10 Track2-Task1, reaching a joint goal accuracy of 0.3605 in the test set.

Introduction

A task-oriented dialogue system interacts with a human in natural language, and completes routine human queries such as hotel bookings, movie ticket purchases, restaurant reservations, and weather queries. It can be used by smart services such as Apple Siri and Xiaodu Assistant, shopping guide robots such as JD Jimi and Ali Xiaomi, or embedded as chat bots and virtual agents in online shopping apps.

Task-oriented dialogue systems usually have two construction methods in the research field. The first is based on the pipeline. This method divides the dialogue system into four cascaded modules: natural language understanding, dialogue state tracking (DST), dialogue strategy, and natural language generation (Young 1999). The user's text input is processed by the four modules in turn, and finally a text response is generated. This method is also the favored processing method in the industry. The other is the end-to-end approach, which trains a neural network to generate output directly according to user input. This approach is still in its infancy (Chen et al. 2017), and isn't as relevant for the time being.

DST is a core component in a pipeline-based task-oriented dialogue system. Because the response of a task-oriented dialogue system depends on its results, excellent DST can improve the user experience by reducing the number of interactions with humans. Its main goal is to maintain the context of the dialogue in the process of human-machine dialogue and display it in the form of a predefined state.

The challenge in DSTC10 Track 2 is to train a multi-domain DST without the use of a large amount of training data, as this challenge track just releases a small amount of Automatic Speech Recognition (ASR) data. To solve the task, we introduced masked span language modeling for DST, and found it to be more efficient in transferring the pre-trained model to downstream tasks. We also explore additional techniques, such as task adaptive multi-task pre-training, task-specific post-processing and data augmentation. The main contributions of the paper are as follows:

- **Masked Span Language Modeling for DST** Following (Raffel et al. 2020), we replace spans of domains and dialogue state slots by a sentinel token(e.g., <extra_id_0>) which is unique to the input sequence. The output sequence that consists of the ground-truth tokens, delimited by the sentinel tokens used to replace them in the input. This is the first time that the masked span language modeling is applied to the dialogue state generation, and brings better generalization capabilities than previous methods. An example of inputs and outputs for masked span language modeling DST is shown in Table 1.
- **Task Adaptive Multi-Task Pretraining** For pretraining of dialogue state generation, we supplemented external topically related datasets (e.g., MultiWoz 2.2, DSTC2/3, official database) to the data provided by the track to design three auxiliary tasks to train the DST. Figure 1 shows the examples of these auxiliary tasks. The experiment results demonstrate that these auxiliary tasks have a positive performance improvement for this track.
- **Task-specific Post-processing** As for spoken conversations recognition, the official ASR engine, which is pre-trained on 960 hours of Librispeech and fine-tuned with speech data, achieved a 24.09% WER compared to the manual transcripts. Because of ASR errors, especially ontology errors, lead to this ASR engine having an incorrect dialogue state. Inspired by (Ren, Ni, and McAuley 2019), we proposed a task-specific heuristic post-processing technique to correct correlated slot-values by using the official database as ground truth information.
- **Data Augmentation** To further improve input diversity,

dialogue	U:m plannin to visit san francisco uh and looking for a zoo S:sure let me see what i can find for you. ok so there is one option called the san francisco zoo is that something that might interest you U:yeah uh can you give me the zip code
input for domain classification	user:m plannin to visit san francisco uh and looking for a zoo system:sure let me see what i can find for you. ok so there is one option called the san francisco zoo is that something that might interest you user:yeah uh can you give me the zip code. We can know that its domains have <extra_id_0>.
output for domain classification	<extra_id_0>attraction
input for dialogue state tracking	user:m plannin to visit san francisco uh and looking for a zoo system:sure let me see what i can find for you. ok so there is one option called the san francisco zoo is that something that might interest you user:yeah uh can you give me the zip code. From domain attraction, we can know that area is <extra_id_0>, name is <extra_id_1>, type is <extra_id_2>.
output for dialogue state tracking	<extra_id_0>None <extra_id_1>San Francisco Zoo <extra_id_2>Zoo

Table 1: An example of inputs and outputs for masked span language modeling dialogue state tracker.

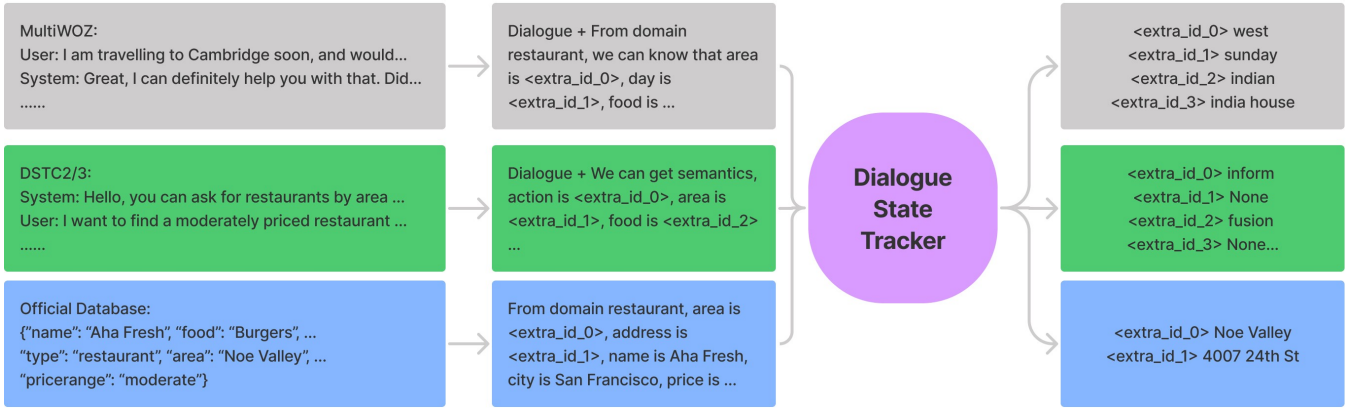

Figure 1: Multi-task pretraining that uses data from 3 different external sources: MultiWOZ 2.2, DSTC2/3, and the official ontology database.

we randomly choose input user utterance from the set of n-best ASR hypotheses and add the slot name of ontology matched by the official database to each utterance.

To facilitate future work on dialogue state tracker, we will publish our training data and source code at **https://github.com/Luka0612/MS-DST**.

Related Work

Dialogue state tracking (DST) in multi-domain task-oriented dialogue systems is an active research area that has existed for more than ten years. Public availability of new large scale datasets, such as MultiWOZ 2.0 and MultiWOZ 2.1 (Budzianowski et al. 2020), stimulate quality research in this field. Due to their large-scale ontology and open vocabulary-based setting, these datasets have been frequently used in training for task-oriented dialogue systems.

Traditional DST approaches rely on either the semantics extracted by the NLU module (Williams 2014) or some hand-crafted features and complex domain-specific lexicons (Rastogi, Gupta, and Hakkani-Tur 2018) to predict the dialogue state. These methods usually suffer from poor scalability and sub-optimal performance. They are also vulnerable to lexical and morphological variations.

Nowadays, there are also many methods based on pre-training models in the DST task. Since the model is pre-trained on a large corpus and computer, it will have very powerful results when transferred to a downstream task. BERT (Devlin et al. 2019) and GPT-2 (Radford et al. 2019) are the most commonly used techniques, and demonstrate leading performance for many downstream tasks in NLP. As a result, the DST research has been shifted to building new models on top of these powerful pre-trained language models. For example, SUMBT (Lee, Lee, and Kim 2019) employs BERT to learn the relationships between slots and dialogue utterances through a slot-word attention mechanism. CHAN (Shan et al. 2020) is established on the basis of SUMBT,considering both slot-word attention and slot-turn attention. TOD-BERT (Wu et al. 2020) further pretrains the original BERT model using several task-oriented dialogue datasets in order to better model dialogue behaviors during the pre-training. SOM-DST (Kim et al. 2020) considers the dialogue state as an explicit fixed-sized memory and selectively overwrites this memory to avoid predicting the dia-

logue state from the beginning in each turn. And the solution of dialogue state is divided into two sub-tasks, namely state operation prediction and slot value generation. TripPy (Heck et al. 2020) uses three copy mechanisms to extract slot values. MinTL (Lin et al. 2020) exploits T5 (Raffel et al. 2020) and BART (Lewis et al. 2019) as the dialogue utterance encoder to jointly learn dialogue states and system responses. It also introduces Levenshtein belief spans to track dialogue states efficiently.

However, properly transferring the pre-trained model knowledge to DST tasks is still controversial, especially as training objectives and the formats of input representations and output representations require different implementations. NP-DST (Ham et al. 2020) and SimpleTOD (Hosseini-Asl et al. 2020) adopt GPT-2 as the dialogue context encoder by combining delimiter tokens, user utterance and system response into input representation and generate dialogue state by auto-regressive training objective. (Feng, Wang, and Li 2021) proposed BERT-based Seq2Seq-DU containing two encoders to respectively encode the utterances in the dialogue and the descriptions of schemas. But the DST training mechanism of the above models is different from the corresponding pre-training mechanism. We are the first to apply the masked span language modeling to the dialogue state generation. (Zhao et al. 2021) also adopted T5 as the backbone and demonstrated pre-training procedures with masked span prediction are more effective than auto-regressive language modeling, but still presented different input and output sequences from pre-training procedures for the DST task.

Approach

As shown in Figure 2, we propose a pipeline composed of 3 discrete units: the domain classifier, the dialogue state tracker, and the post-processing unit. The output of a previous unit is fed into the next unit as input, ultimately leading to a JSON object that represents the current dialogue state. The domain classifier and the dialogue state tracker are both encoder-decoder transformer models trained using customized tasks, and the post-processing unit is a series of heuristic methods based on the official ontology database.

As noted in the related work section and the DSTC10 Track II proposal, there have been more public datasets and well-performing models available for task-oriented dialogues (Kim et al. 2021). Most of these models are trained on written conversations, which differ noticeably from spoken conversations. Given the same context, intention, and semantics, spoken dialogues often have more noises and errors from audio capture, speech recognition, and oral speech inconsistencies.

Since the official development data only contains ASR outputs from 100 conversations with considerable noises and errors, the critical challenge is to effectively leverage available task-oriented dialogue datasets and the official development data. To address this challenge, we propose a series of pre-training and fine-tuning tasks, inspired by the unsupervised training objective employed by T5 model (Raffel et al. 2020).

We train two T5 models as backbones of our domain classifier and dialogue state tracker. As suggested by its name "Text-to-Text Transfer Transformer", the model has an encoder-decoder transformer architecture similar to the original transformer (Vaswani et al. 2017). T5 is trained by reformatting a wide range of NLP tasks into text-to-text formats. The most notable characteristic of the model is its outstanding ability in transfer learning.

DST Training Objective

Following the publication of BERT (Devlin et al. 2018), large scale pre-trained language models utilizing the masked language modeling paradigm have taken the NLP world by storm. Despite the good performance in subsequent tasks, BERT's masked language modeling objective presents several drawbacks: reproducing the original text without corruption in the output is computationally demanding; the use of [MASK] tokens in the pre-training inputs and the lack of [MASK] tokens in the fine-tuning inputs create a misalignment, which may lead to a sub-optimal performance in transfer learning.

To address these drawbacks for DST, we introduce a novel training methodology by integrating the span-based masking strategy used in the unsupervised pre-training of T5 (Raffel et al. 2020).

For the input sequence, the span-based masking strategy use a unique sentinel token <extra_id_n>to randomly mask a span of text (one or more continuous tokens). The output sequence is formulated by concatenating masked sequences, each prepended with its corresponding sentinel token. Span-based masking strategy has been shown to exhibit superior performance compared to the BERT-like masking, where only every masked token is replaced by one general masking token [MASK] (Joshi et al. 2019).

To apply this span-based masked language modeling paradigm in supervised training, we further add a prompt to indicate the task for the model. As shown in Table 1, the prompt contains unique sentinel tokens to instruct the model to predict domains or other dialogue states. This formulation of supervised training reduces the computational complexity by having the model generate masked spans, and it also unifies the supervised and unsupervised training tasks. This supervised training method is used throughout the series of pre-training and fine-tuning tasks we devised to train our domain classifier and DST. To differentiate between the BERT-like masked language modeling paradigm and our span-based masked language modeling paradigm, we will refer to the latter as masked span language modeling (MSLM) paradigm.

Domain Classification

Each dialogue in the development data corresponds to one or more of 3 domains: "restaurant", "hotel", and "attraction". Knowing the domain associated with each dialogue query reduces the space of possible dialogue states, so the first step in the pipeline classifies each dialogue to its corresponding domain.

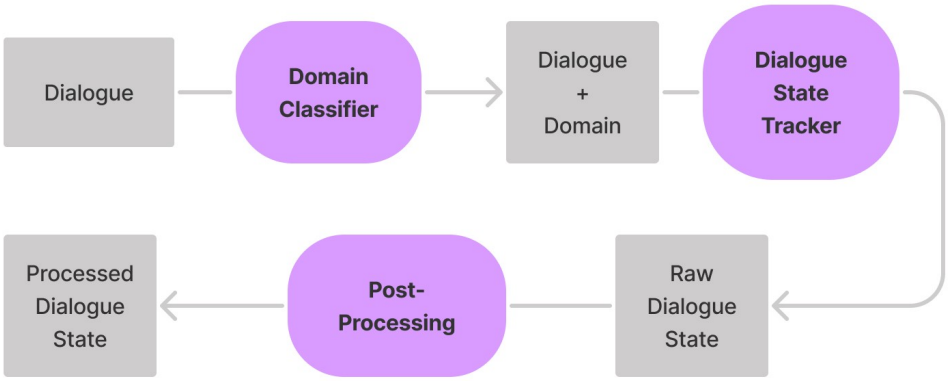

Figure 2: The 3-unit pipeline that extracts dialogue states. Domain classifier and dialogue state tracker are two transformer models, and the post-processing unit is a series of heuristics methods.

Pre-Training The dialogue format and semantics of MultiWOZ 2.2 are similar to those of the development data, so we use MultiWOZ for pre-training to improve model generalizability (Budzianowski et al. 2020). The development data most notably differs from MultiWOZ in that the dialogue state labels are provided dialogue by dialogue, rather than turn by turn. To simulate the format of the development data, we first concatenate all utterances between the system and the user for each dialogue and aggregate the domains covered by all the utterances.

To apply the MSLM paradigm, we append the prompt "We can know that its domains have <extra_id_0>" to every concatenated dialogue string as the input, and we then format the aggregated domain as "<extra_id_0>attraction, restaurant", which acts as a label. The MSLM objective is consistent with the unsupervised pre-training objective of the T5 model. The consistency aims to maximize the T5's transferred performance when learning to predict the domain.

Fine-Tuning The pre-trained model is then trained on the development data. We employed the same MSLM objective design by providing masked prompts and using ground truth domain(s) as training labels.

Dialogue State Tracking (DST)

Similar to training the domain classifier, the training of the dialogue state tracker is divided into 2 steps: pre-training on external datasets and fine-tuning on the development data. Both steps are trained based on the MSLM paradigm.

Pre-Training We developed 3 additional pretraining tasks (Figure 1) to augment the small training data in the task. Each task targets a different ability necessary to accurately track dialogue states. We employed a multi-task strategy and trained all 3 tasks together.

The first task uses MultiWOZ data, which contains over 10000 dialogues, and it aims to improve the model's generalizability in extracting states from diverse utterances (Budzianowski et al. 2020). As shown in the first row in Figure 1, the concatenated MultiWOZ dialogue is appended with a domain-dependent prompt. For example, for a dialogue in the domain of "restaurant", the appended prompt would be "From domain restaurant, we can know that ...", followed by masked statements about states associated with the current domain such as "area is <extra_id_0>" and "day is <extra_id_1>". In these statements, only the values were masked by the sentinel tokens. Following the MSLM paradigm, the masked true values were used to generate the label. A typical label looks like "<extra_id_0>centre <extra_id_1>Saturday <extra_id_2>None ... <extra_id_6>11:15".

The second task uses DSTC2/3 data, whose dialogues contain spoken user utterances (Henderson, Thomson, and Williams 2014a,b). These spoken dialogues were used to improve the model's ability to extract correct dialogue states from noisy ASR transcripts of spoken utterances. As illustrated by the middle row in Figure 2, the task also follows the MSLM paradigm. The input data format is slightly different from the first task: the prompt is formulated as "We can get semantics", followed by masked states as outlined in the first task. All in all, this label resembles the label in the first task.

The third task uses the official ontology database, where each entry lists information about an attraction, a restaurant, or a hotel. The task aims to infuse the model with relevant ontological knowledge, so that it can better predict the target states given incomplete or noisy data. For example, assuming the model has learned that Aha Fresh is a restaurant at Noe Valley that serves burgers, an input conversation that mentions the search for a burger restaurant at Noe Valley may boost the likelihood of the "name" of the restaurant being "Aha Fresh", which may not be correctly captured in the ASR transcript. As shown in the bottom row of Figure 1, the input data starts with a prefix of "From domain attraction/hotel/restaurant," followed by statements of dialogue states of the attraction/hotel/restaurant, which is similarly masked as the first two tasks. However, the properties are randomly, rather than all, masked, and the values masked were used to generate MSLM labels, effectively guiding the

model to predict the masked values.

Fine-Tuning The official development data is formatted in the same manner as mentioned in task 1 of pre-training: adding domain-dependent prompt as input and using corresponding true values as labels. To increase the variety of the inputs, user utterances were sampled from the 10-best ASR transcripts provided. And we further leverage the official ontology database by randomly providing hints in dialogues of what dialogue state a word possibly represents. For example, "San Francisco" in the dialogue is appended with "(city)" to indicate the possible dialogue state it represents.

Post-Processing

ASR errors often propagate to an incorrect ontological prediction, and we have come up with three heuristics to remove some of these errors via post-processing. The first heuristic resembles auto-correct: if the prediction doesn't match any value in the corresponding ontology database, but the Levenshtein distance ratio between our prediction and the closest candidate in the provided ontology database is over 0.5, then the prediction is replaced by the closest candidate. The second heuristic removes the prediction if it does not match any candidates in the above procedure. The third heuristic deals with the problem where the model predicts incompatible states as if they belong to 2 items in the database. The method utilizes a voting mechanism, where the item with a higher number of matching terms wins. The losing item's related dialogue states are then removed.

Experimental Results

We use T5-3b as the backbone architecture to generate the domains and then to generate a final dialogue state for every domain.

Datasets

Dataset for Pre-Training To generate dialogue states, we have used several dialogue corpora to pre-train on a multi-task mixture before fine-tuning on the development data provided by DSTC10. MultiWOZ 2.2 is an enhanced version of MultiWOZ 2.1 that fixes dialogue state annotation errors and redefines the ontology by disallowing the vocabularies of slots. In order to reduce the negative impact of irrelevant domains, we only use the domains restaurant, hotel, attraction. However, MultiWOZ 2.2 includes only written conversations which differ from spoken conversations in this challenge. Therefore, we add DSTC2/3 datasets to focus on spoken dialogue (Henderson, Thomson, and Williams 2014a,b). Meanwhile, we have converted database entry into a text-to-text format and randomly masked less than four spans of text by so-called sentinel tokens for Commonsense Reasoning. We show data statistics in Table 2. For domain generation, only MultiWOZ 2.2 is used for pre-training.

Dataset for Fine-Tuning The official dataset for DSTC10 track 2 contains 100 dialogues for development and 700 dialogues for testing. And the development/testing data includes 936/6588 instances each of which is a partial conversation from the beginning to the target user turn, respectively. As there was no large-scale training data released for this challenge track, we use the development data for the model fine-tuning.

Name	Examples	task
MultiWoz 2.2	60206	DST
dstc 2/3	35784	SLU
track DB	2565	CR

Table 2: Statistics of dialog corpora for pretraining. DST: dialogue state tracking, SLU: spoken language undestanding, CR: Commonsense Reasoning

Data Augmentation As the DSTC10 validation dataset contains the n-best ASR outputs for spoken conversations, we randomly and uniformly choose input user utterances from the set of 10 ASR hypotheses. In addition, we add the slot name of ontologies matched by the official database to each utterance (e.g. "San Francisco" $\rightarrow$ "San Francisco(city)"), which hints this ontology may be a dialogue state value. Through these two methods, we obtain a set of 3 alternatives (including the original one) for each instance, which effectively improves the diversity of fine-tuning data.

Training Details

We developed our model using Huggingface (Wolf et al. 2020) T5-3b that consists of a 24 layers encoder and decoder. It has 2.8 billion parameters with 1024 layer size and 32-headed attention, and is pre-trained on "Colossal Clean Crawled Corpus".

For the generation of dialogue state, we pre-trained T5-3b on the multi-task mixture for 3000 steps on a batch size of 48 sequences of length 1024 and fine-tuned on the official development data with batch size 24 for 2400 steps. For domain generation, we pre-trained and fine-tuned for 300 steps and 150 steps with batch size 24, respectively. The optimizer was Adam (Kingma and Ba 2015) with learning rate of 5e-5.

All output sequences were generated using greedy decoding. Also as (Holtzman et al. 2020), We experimented with beam search, but observed no improvement.

Automatic Evaluation

The following metrics were used to evaluate the performance in the first sub-track of Track 2 in DSTC10: Joint Goal Accuracy, Slot Accuracy, Value Precision/Recall/F-measure, None Precision/Recall/F-measure. The Joint Goal Accuracy computes the percentage of the predicted dialog states which are exactly equal to the ground truth. Table 3 shows the official ranking based on the joint goal accuracy.

We rank in second place, reaching a joint goal accuracy of 0.3605 in the test set. Team A11 used an ensemble model that combined multiple models based on BERT Large and PLATO-2 to generate the final results. Our approach is ranked first from a single model perspective.

Ablation Study

Since the development data was used for model fine-tuneing, we only performed a more detailed ablation study of the pro-

Rank	Team	JGA	SA	Value P/R/F	None P/R/F	Ensemble
1	A11	46.16	94.98	92.24/90.09/91.15	97.07/98.58/97.82	Y
2	A01(ours)	36.05	93.67	92.08/86.71/89.31	94.91/98.61/96.72	N
3	A07	27.73	89.48	81.54/77.56/79.50	94.64/97.73/96.16	Y
N/A	Baseline	0.39	70.52	61.30/30.97/41.15	73.02/97.16/83.38	N

Table 3: Official results of the evaluation. The rank is based on the joint goal accuracy. JGA: joint goal accuracy, SA: slot accuracy, Value P/R/F: value precision/recall/f-measure, None P/R/F: none precision/recall/f-measure, Ensemble:whether to combine multiple system outputs into one entry

Method	Joint Goal Accuracy
T5-3B	22.22
+MSLM	24.59
+MultiTask Pretrain	25.93
+Data Augmentation	26.08
+Postprocess	36.05

Table 4: The ablation study of dialogue state generation

posed components on the test data. The results are shown in Table 4.

The experiment results show that all proposed contributions have a positive effect on system performance. We can see that the performance of masked span language modeling (MSLM) for DST is further improved by more than 2.37%, compared with auto-regressive DST. This demonstrates that masked span prediction based on T5 is more effective for dialogue state tracking.

In addition, we find that heuristic post processing achieves surprisingly large performance improvement. Through further analysis, it is found that the performance improvement is mainly derived from the first heuristic (+7.24 JGA): a simple ontology replacement by the closest Levenshtein distance candidate, especially ontology for name and area slots. Although masked span prediction correctly locates the slot-value in the dialogue, it does not correct the wrong ASR ontology because vanilla T5 is trained on written data and the finetune data including the ASR outputs of this challenge conversations is small (100 dialogues). In order to better solve this problem, pre-training a self-supervised language model on spoken language data is our future work.

Conclusions and Future Works

In this paper, we discuss our findings when a dialogue state tracker based on masked span language modeling (MSLM) was introduced as the solution for DSTC10 Track2-Task1 competition. MSLM can transfer knowledge better from the pre-training stage to the dialogue state tracking tasks. We also devised three auxiliary tasks to improve model generalizability and used task-specific heuristic post-processing to correct correlated slot-values. According to our ablation study, MSLM paradigm and post-processing contribute most to our system's performance. Our proposed model is the best performing single model and is ranked second in the overall objective evaluation of DSTC10 Track2-Task1, in term of the value prediction precision metric.

In the future, we plan to focus on language representation for spoken conversations which are more complex than written conversations and have more errors in ASR. However, most language models are pre-trained based on written texts. We will pre-train our self-supervised language model from scratch on a large amount of spoken data.

Acknowledgement

The authors acknowledge Katherine M. Li and Jack P. Li for their assistance in proofreading the paper.

References

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gašić, M. 2020. MultiWOZ – A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. arXiv:1810.00278.

Chen, H.; Liu, X.; Yin, D.; and Tang, J. 2017. A Survey on Dialogue Systems. *ACM SIGKDD Explorations Newsletter*, 19(2): 25–35.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Feng, Y.; Wang, Y.; and Li, H. 2021. A Sequence-to-Sequence Approach to Dialogue State Tracking. arXiv:2011.09553.

Ham, D.; Lee, J.-G.; Jang, Y.; and Kim, K.-E. 2020. End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 583–592. Online: Association for Computational Linguistics.

Heck, M.; van Niekerk, C.; Lubis, N.; Geishauser, C.; Lin, H.-C.; Moresi, M.; and Gašić, M. 2020. TripPy: A Triple Copy Strategy for Value Independent Neural Dialog State Tracking. arXiv:2005.02877.

Henderson, M.; Thomson, B.; and Williams, J. D. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, 263–272.

Henderson, M.; Thomson, B.; and Williams, J. D. 2014b. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, 324–329. IEEE.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The Curious Case of Neural Text Degeneration. arXiv:1904.09751.

Hosseini-Asl, E.; McCann, B.; Wu, C.-S.; Yavuz, S.; and Socher, R. 2020. A Simple Language Model for Task-Oriented Dialogue. arXiv:2005.00796.

Joshi, M.; Chen, D.; Liu, Y.; Weld, D. S.; Zettlemoyer, L.; and Levy, O. 2019. SpanBERT: Improving Pre-training by Representing and Predicting Spans. arXiv:1904.10529.

Kim, S.; Liu, Y.; Jin, D.; Papangelis, A.; Hedayatnia, B.; Gopalakrishnan, K.; and Hakkani-Tur, D. 2021. DSTC10 Track Proposal: Knowledge-grounded Task-oriented Dialogue Modeling on Spoken Conversations.

Kim, S.; Yang, S.; Kim, G.; and Lee, S.-W. 2020. Efficient Dialogue State Tracking by Selectively Overwriting Memory. arXiv:1911.03906.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lee, H.; Lee, J.; and Kim, T.-Y. 2019. SUMBT: Slot-Utterance Matching for Universal and Scalable Belief Tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5478–5483. Florence, Italy: Association for Computational Linguistics.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv:1910.13461.

Lin, Z.; Madotto, A.; Winata, G. I.; and Fung, P. 2020. MinTL: Minimalist Transfer Learning for Task-Oriented Dialogue Systems. arXiv:2009.12005.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Rastogi, A.; Gupta, R.; and Hakkani-Tur, D. 2018. Multi-task Learning for Joint Language Understanding and Dialogue State Tracking. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 376–384. Melbourne, Australia: Association for Computational Linguistics.

Ren, L.; Ni, J.; and McAuley, J. 2019. Scalable and Accurate Dialogue State Tracking via Hierarchical Sequence Generation. arXiv:1909.00754.

Shan, Y.; Li, Z.; Zhang, J.; Meng, F.; Feng, Y.; Niu, C.; and Zhou, J. 2020. A Contextual Hierarchical Attention Network with Adaptive Objective for Dialogue State Tracking. arXiv:2006.01554.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Łukasz Kaiser; and Polosukhin, I. 2017. Attention Is All You Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*. Online: Conference on Neural Information Processing Systems.

Williams, J. D. 2014. Web-style ranking and SLU combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 282–291. Philadelphia, PA, U.S.A.: Association for Computational Linguistics.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Wu, C.-S.; Hoi, S.; Socher, R.; and Xiong, C. 2020. TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue. arXiv:2004.06871.

Young, S. 1999. Probabilistic Methods in Spoken Dialogue Systems. *Philosophical Transactions of the Royal Society (Series A*, 358: 1389–1402.

Zhao, J.; Mahdieh, M.; Zhang, Y.; Cao, Y.; and Wu, Y. 2021. Effective Sequence-to-Sequence Dialogue State Tracking. arXiv:2108.13990.