End-to-End Neural Pipeline for Goal-Oriented Dialogue System using GPT-2

Donghoon Ham,[1*] **Jeong-Gwan Lee,**[1*] **Youngsoo Jang,**[1] **Kee-Eung Kim**[1,2]
[1]School of Computing, KAIST, Daejeon, Republic of Korea
[2]Graduate School of AI, KAIST, Daejeon, Republic of Korea
{dhham, jglee, ysjang}@ai.kaist.ac.kr, kekim@kaist.ac.kr

Abstract

The first sub-task in the multi-domain task-completion dialogue challenge track in the 8th dialogue systems technology challenge (DSTC8) requires participants to build an end-to-end dialogue system that is capable of complex multi-domain dialogues. The traditional approach to build such a dialogue system is to take a pipelined architecture, where its modular components are optimized individually. However, such an optimization scheme does not necessarily yield the overall performance improvement of the whole system. On the other hand, most end-to-end dialogue systems with monolithic neural architecture are trained only with input-output utterances, without taking into account the entire annotations available in the corpus. This scheme makes it difficult for goal-oriented dialogues where the system needs to interact with external systems such as database engines or to provide interpretable information about why the system decided to generate a particular response. In this paper, we present an end-to-end neural architecture for dialogue systems that addresses both challenges above. In the official human evaluation, our dialogue system achieved the success rate of 68.32%, the language understanding score of 4.149, and the response appropriateness score of 4.287, which ranked the system at the top position in all performance evaluation criteria.

Introduction

The goal-oriented dialogue systems help users achieve their goal such as requesting information or executing commands via natural language conversations. It is crucial for the dialogue system to keep track of the dialogue flow and carry out an effective conversation, even when the user goal is complicated or dialogue flow is suddenly changed.

The traditional approach to building a goal-oriented dialogue system mostly adopts a pipelined modular architecture, with the natural language understanding (NLU) module (Kim, Lee, and Stratos 2017; Lee et al. 2019) that first recognizes and comprehends user's intent and extracts values for slots, then the dialogue state tracking (DST) module (Williams et al. 2013) that tracks the values of slots, then the dialogue policy (POL) module that decides the system action, and then finally the natural language generation (NLG) module (Wen et al. 2015) that generates the utterance that corresponds to the system action. In some cases, multiple modules are combined together, e.g. the Word-level DST (Ramadan, Budzianowski, and Gašić 2018; Wu et al. 2019; Lee, Lee, and Kim 2019) which maps a dialogue history into dialogue state as a composite of NLU as DST, and the Word-level POL (Budzianowski et al. 2018; Pei, Ren, and de Rijke 2019; Chen et al. 2019; Mehri, Srinivasan, and Eskenazi 2019; Zhao, Xie, and Eskenazi 2019) which maps a previous utterance and dialogue state into the system response as a composite of POL and NLG.

These modules are usually optimized separately, which does not necessarily lead to overall performance optimization for successful task completion. On the other hand, end-to-end neural models for dialogue systems (Madotto, Wu, and Fung 2018; Lei et al. 2018) enjoy a straightforward training approach to generating system responses, but it is difficult for goal-oriented dialogues where the system needs to interact with external systems or to generate an explanation that supports why the system generated a particular response.

The first sub-task in the multi-domain dialogue challenge in DSTC8 required participants to build an end-to-end dialogue system for tourist information desk settings involving complex multi-domain dialogues. The objective of the system is to trace the user's requirement, provide correct information that the user requests, and optionally make an appropriate booking using an external database. The dialogue system should be based on the MultiWOZ dataset (Budzianowski et al. 2018), which is a collection of human-human conversations covering multiple domains and topics enriched with annotation to facilitate machine learning approaches to build the system. To support this, ConvLab (Lee et al. 2019) was released to serve as a development platform, supplied with reusable component models in pipelined dialogue systems as well as end-to-end models, supporting various aspects in the development and evaluation phases of dialogue systems.

Our dialogue system for the competition is based on an end-to-end *monolithic* neural network based on GPT-2 (Rad-

[*]These authors contributed equally.

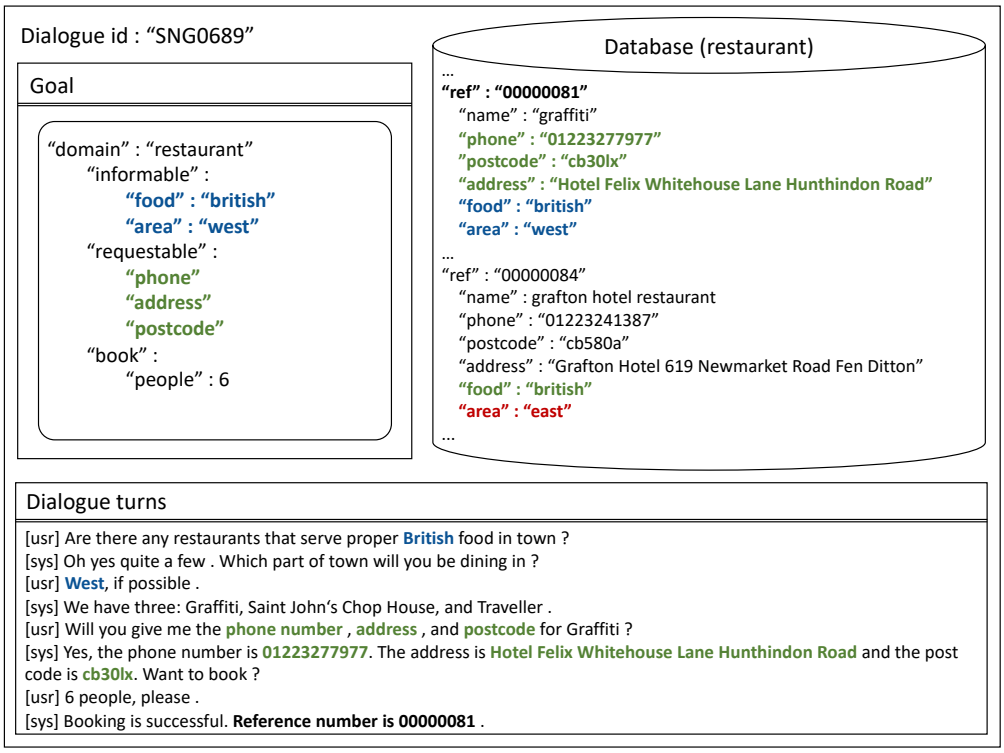

Figure 1: A single-domain example in MultiWOZ dataset.

ford et al. 2019), which performs the following essential dialogue management steps in a sequential manner: (1) DST via predicting the dialogue state, (2) POL via predicting the system action, (3) retrieving appropriate records from the external database for the dialogue state and the policy, and (4) NLG via predicting the next response. As a result, our neural model not only generates the response utterance as in any end-to-end neural dialogue systems but also generates word-level dialogue states and word-level dialogue policies as intermediate outputs improving the interpretability of the behavior of the dialogue system. The rich annotation of dialogue states and dialogue policies provided in the MultiWOZ dataset allowed us to train our system in a very natural way.

The evaluation process involved both automatic and human evaluations. In the automatic evaluation using a user simulator, our system attained success rate of 79.40% and F1 score of 0.83, ranking at the 5th place. However, in the human evaluation using crowd human workers, our system attained success rate of 68.32%, language understanding score of 4.149, and response appropriateness score of 4.287, ranking at the 1st place. We suspect the relatively low performance in the automatic evaluation is due to the imperfect user simulator, which is inevitable in any automatic evaluation involving simulations.

The main characteristics of our system can be summarized as follows: (1) it is trained to follow the traditional pipelined process in dialogue systems, making the monolithic neural model more interpretable and easily integrable with external systems, while (2) it is trained in an end-to-end fashion with gradient descent, and (3) leverages GPT-2, a powerful pre-trained language model.

Task Description

End-to-End Multi-Domain Task-Completion Task

In this paper, we focus on the first sub-task of DSTC8, end-to-end multi-domain task-completion dialogue. The goal of this challenge is to build a dialogue system for tourist information desk settings covering various domains. This task is based on the multi-domain MultiWOZ dataset (Budzianowski et al. 2018) and the ConvLab platform (Lee et al. 2019).

The MultiWOZ Dataset

The MultiWOZ dataset is a fully annotated corpus for goal-oriented dialogue system development. Each dialogue is rich in annotations such as 'goal', 'metadata', and 'dialog act' as well as user and system utterances. These annotations allow us to train and validate individual components of a pipelined dialogue system (NLU, DST, POL, NLG) or an end-to-end dialogue system using supervised learning.

Figure 1 shows an example of a single-domain dialogue in the MultiWOZ dataset. Each dialogue consists of 'Goal', 'Database' and 'Dialogue turns'. The goal is defined by the

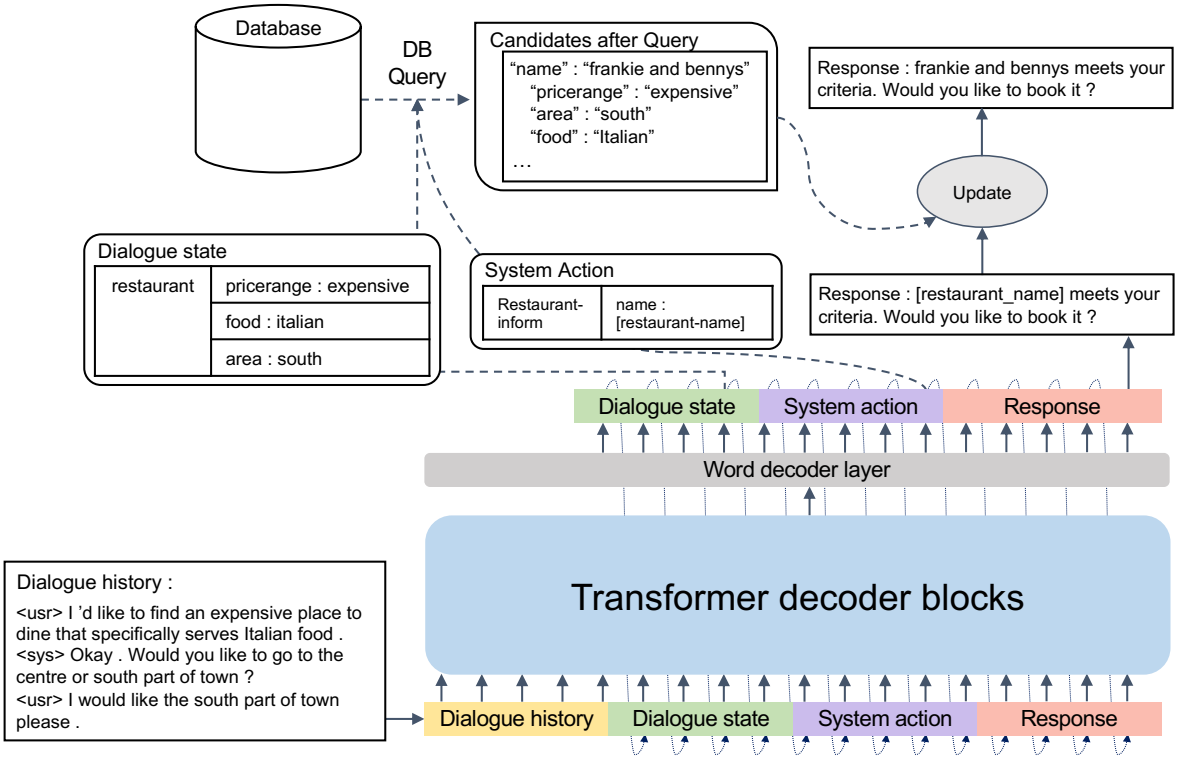

Figure 2: The overview of our end-to-end neural dialogue model fine-tuned on GPT-2. The dashed line represents the information to and from the DB query, which is invoked when the system action needs to fetch an actual value from the database.

domain and the slots. The slots are divided into *informable*, *requestable* and *book* slots. *Informable* slots represent a user constraint and *Requestable* slots indicate additional information that a user want to obtain through a request. *book* slots allow the user to reserve a place recommended by the system. The system should help the user to book and obtain information based on what the user has mentioned.

ConvLab

In addition, a development platform is provided to support participants. ConvLab (Lee et al. 2019) is an open-source platform that supports researchers to train and evaluate their own dialogue systems. ConvLab contains the state-of-the-art models of NLU, DST, POL, NLG (Kim, Lee, and Stratos 2017; Lee et al. 2019; Ramadan, Budzianowski, and Gašić 2018; Wu et al. 2019; Wen et al. 2015; 2017; Budzianowski et al. 2018) and an end-to-end neural model for dialogue systems (Lei et al. 2018; Madotto, Wu, and Fung 2018), which are reusable for building dialogue systems using various approaches.

ConvLab also provides an agenda-based user simulator consisting of a multi-intent language understanding (MILU) module, rule-based policy, and template-based NLG module. For each dialogue, a goal is generated that conforms with the goal schema of the MultiWOZ dataset. The user simulator then generates an agenda based on the goal. While interacting with a dialogue system model, it recognizes the system dialogue acts, decides the user policy from the agenda stack, and generates the next response at each turn. When the system offers to book and the user accepts that booking, the system should notify that the booking is completed with an 8-digit reference number. The reference number is used to verify whether the booking is correct. Although the user simulator is highly sophisticated, it is not as perfect as a human. Hence, the dialogue systems submitted to the challenge were evaluated not only with the user simulator but also with crowdsourced human users.

End-to-End Neural Pipeline for Goal-Oriented Dialogue System

In this section, we describe our end-to-end neural pipeline for goal-oriented the dialogue system based on GPT-2. Our model consists of (1) the finetuned GPT-2 model and (2) the database query module. We take the pre-trained GPT-2 model and finetune it to follow the steps in the dialogue management pipeline. Figure 2 illustrates an overall architecture with a concrete example of the process. The overview of the process of our model as follows:

1. Generate a dialogue state conditioned on the dialogue history.
2. Generate a system action with delexicalized tokens con-

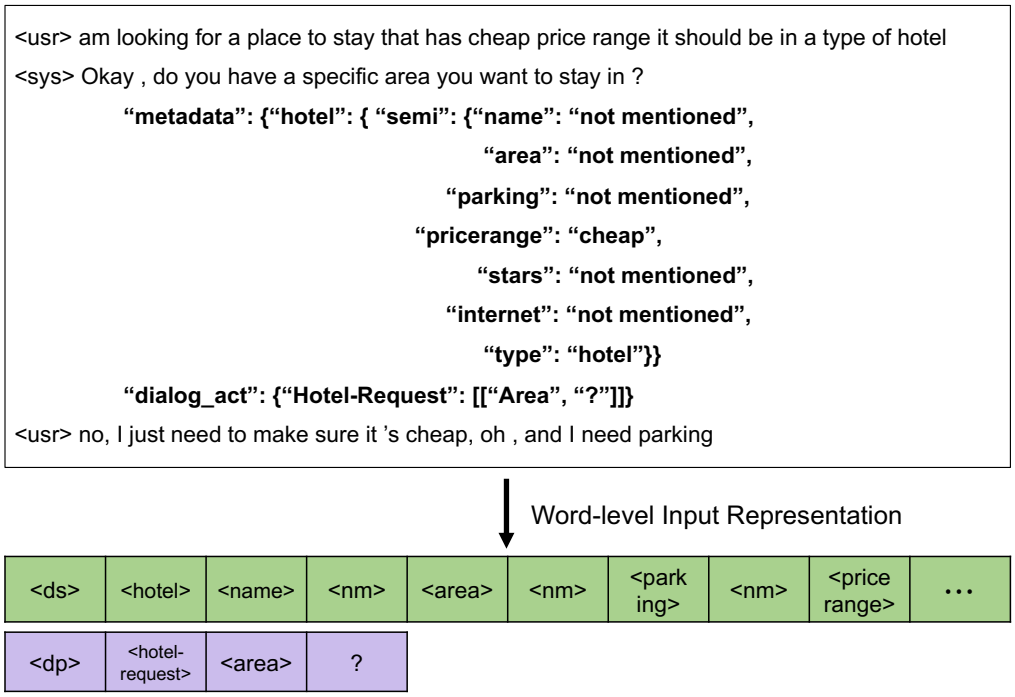

Figure 3: The rich annotations of the MultiWOZ dataset, the 'metadata' treated as dialogue state traced until current user turn and 'dialogue act' treated as the system action are placed on the system's turn.

ditioned on the dialogue history and dialogue state.[1]

3. If the generated system action (e.g. 'inform', 'book') needs external information from the database, the query module[2] retrieves the candidates and returns one of them.
4. Update the current system action to match attributes of candidates obtained in step 3.
5. Generate the system response with delexicalized tokens conditioned on the dialogue history, dialogue state, and updated system action.
6. Based on the selected candidate from step 3, update the delexicalized tokens in response and generate the final system response.

We use the rich set of annotations from the MultiWOZ dataset to fine-tune the GPT-2, turning it into a complete dialogue system that exactly follows the steps in the above process.

Input Representation

In MultiWOZ dataset, 'metadata' records the current dialogue state and 'dialog_act' represents the current system action (see Figure 3). Since the only text input can be operated with GPT-2, the dialogue state and system action should be converted to word-level tokens.

Figure 3 shows an illustrative example of the single-turn in single-domain dialogue and word-level representation of its dialogue state and system action. The domain and the slot names are represented by additional special tokens, because they perform a special role in whole process of dialogue system. `<nm>` and `<dc>` are special tokens that indicate '*not mentioned*' and '*don't care*'. Also, we introduce delimiter tokens not only `<user>` and `<system>` but also `<ds>` and `<sa>` to give a signal for which segment each word belongs to.

The complete input representation for our model is illustrated in Figure 4, followed by Radford et al. (2018) and Wolf et al. (2019). The input embedding comprises of the token embedding, the speaker embedding, and the positional embedding.

Delexicalization

In the goal-oriented dialogue system, it is a very typical problem to detect out-of-vocabulary (OOV) values. Specifically, particular values such as reference number, postcode, and address are examples of OOV values and database-dependent. For GPT-2, tokenization using subword segmentation (Sennrich, Haddow, and Birch 2016) solves the OOV problem, but at the same time, sampling-based subword decoding leads to generate an imperfect word for low-frequency value of slots, and database-dependent issue still remains. To address those problems, we delexicalized all values of requestable slots (reference number, name, postcode, phone number, address) as `[DOMAIN_SLOTNAME]` in

[1] Delexicalization is the replacement of requestable slot values such as postcodes, name, phone number into generic slot tokens.

[2] ConvLab provides a DB query module returning candidates given domain and dialogue state.

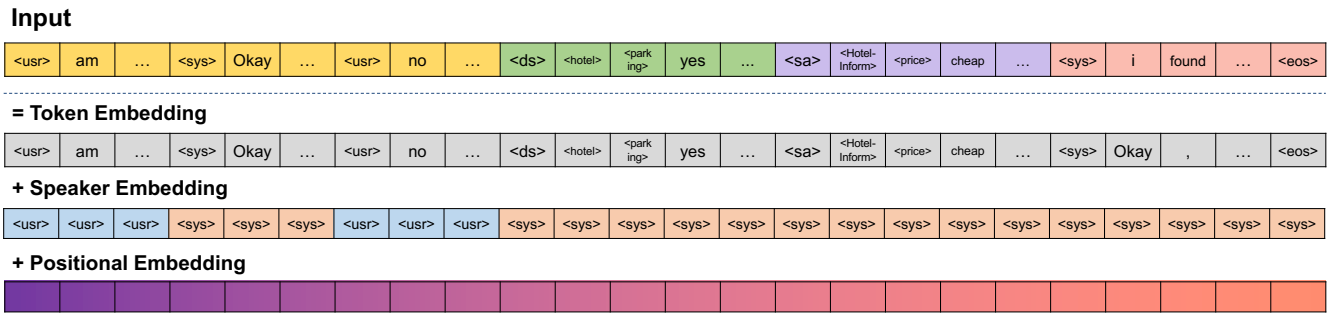

Figure 4: The input representation for fine-tuning phase.

MultiWOZ dataset. Thus, our model finally generates delexicalized system response, and delexicalized tokens are manually replaced by real information from the database.

Multi-task Learning

During the fine-tuning phase, we optimize the weighted sum of the objectives of the language modeling (LM) and the next-utterance classification (NC) (Radford et al. 2018). For the LM, we use the standard left-to-right LM objective (Bengio et al. 2003) as follows:

$$L_{LM}(w_1, \ldots, w_n) = \sum_i \log P(w_i | w_1, \ldots, w_{i-1})$$

The LM objective estimates the likelihood of next word from given previous words.

In NC, the model needs to distinguish the gold utterance *(gold dialogue state+gold system action+gold system response)* from a distractor *(gold dialogue state+gold system action+fake system response)* with given dialogue history. We randomly sample a fake system response from MultiWOZ dataset. The linear classifier takes the last hidden state of GPT-2's transformer decoder block as input and computes the class probability by passing through softmax layer. The cross-entropy loss between class probability and correct label used for the NC objective, L_{NC}. Thus, for given word sequence $W = (w_1, \ldots, w_n)$, the total objective becomes combination of L_{LM} and L_{NC} with weights α_{LM} and α_{NC}:

$$L_{total}(W) = \alpha_{LM} L_{LM}(W) + \alpha_{NC} L_{NC}(W)$$

Decoding Strategy

When we generate the system response from the dialogue history, the final output is the probability distribution of each position overall words. Based on these probability distributions, there are many methods for decoding to word tokens which have a significant impact on the quality of generated output (Holtzman et al. 2019; Weston, Dinan, and Miller 2018). The greedy decoding and beam search are the most common approaches. However, since the greedy decoding only considers the highest probability token, words with high probability that come afterwords with low probability are often not selected. Also, Holtzman et al. (2019) evidences that the beam search decoding is not appropriate for high-entropy natural language generation such as dialogues. Other sampling-based decoding methods, *top-k sampling* and *top-p sampling* as known as *nucleus sampling* (Holtzman et al. 2019) have addressed the above problems quite effectively for dialogue tasks (Wolf et al. 2019; Budzianowski and Vulić 2019). We evaluated the performance of our models with different decoding schemes as mentioned above and selected the best one via human evaluation.

Handling Empty Query Results

As we mentioned before, GPT-2 invokes the query module to interact with the database. However, GPT-2 doesn't know how many candidates satisfy the constraints a-priori. Therefore, there exist cases where no candidate happen to satisfy the constraints, which we refer to as *Empty-Query-Result*. In this case, the dialogue system should generate the system response corresponding to the intent *Empty-Query-Result*. Our system monitors the system action generated from GPT-2 and replace it by `<EQR>` if the database query returns an empty result, and feed this modified input to GPT-2 to generate the system response. This simple solution worked quite well in practice.

Related Work

TransferTransfo (Wolf et al. 2018) was the first attempt to incorporate a large-scale pre-trained language model into a chit-chat dialogue system. Using GPT as a backbone, their fine-tuning approach ranked first in the automatic evaluation and second in human evaluation in the ConvAI2 competition (Dinan et al. 2018). Our model is mainly inspired by this work, extending to goal-oriented dialogues using GPT-2.

Parallel and independent to our work, Budzianowski and Vulić (2019) also demonstrated a goal-oriented dialogue system model using the fine-tuning approach on the MultiWOZ dataset. However, they only handle dialogue-context-to-text task, which outputs the system response given the dialogue history, the oracle dialogue state, and the database. In our case, no oracle information related to database and dialogue state is provided, only the dialogue history is given. Taking dialogue history as an input, our model operates as a complete dialogue system that generates system response by sequentially following the core steps in the dialogue management pipeline.

Rank	Team ID	Success Rate ↑	Return ↑	Turns ↓	Precision ↑	Recall ↑	F1 ↑	Book Rate ↑
1	504429	**88.80%**	61.56	7.00	**0.92**	**0.96**	**0.93**	93.75%
2	504563	88.60%	**61.63**	6.69	0.83	0.94	0.87	**96.39%**
3	504651	82.20%	54.09	**6.55**	0.71	0.92	0.78	94.56%
4	504641	80.60%	51.51	7.21	0.78	0.89	0.81	86.45%
5	**Ours**(504430)	79.40%	49.69	7.59	0.80	0.89	0.83	87.02%
6	504529	58.00%	23.70	7.90	0.61	0.73	0.64	75.71%
7	504666	56.60%	20.14	9.78	0.68	0.77	0.70	58.63%
8	504502	55.20%	17.18	11.06	0.73	0.74	0.71	71.87%
9	504524	54.00%	17.15	9.65	0.66	0.76	0.69	72.42%
10	504569	52.20%	15.81	8.83	0.46	0.75	0.54	76.38%
11	504582	34.80%	-6.39	10.15	0.65	0.75	0.68	N/A
12	504632	00.00%	-58.88	20.88	0.00	0.01	0.00	N/A
N/A	Baseline	63.40%	30.41	7.67	0.72	0.83	0.75	86.37%

Table 1: Overall results of the automatic evaluation. Bold indicates the best score for each metric.

Rank	Team ID	Success Rate ↑	Language Understanding ↑	Response Appropriateness ↑	Turns ↓
1	**Ours**(504430)	**68.32%**	**4.149**	**4.287**	19.507
2	504429	65.81%	3.538	3.632	15.481
3	504563	65.09%	3.538	3.840	**13.884**
4	504651	64.10%	3.547	3.829	16.906
5	504641	62.91%	3.742	3.815	14.968
6	504569	54.90%	3.784	3.824	14.107
7	504529	43.56%	3.554	3.446	21.818
8	504582	36.45%	2.944	3.103	21.128
9	504666	25.77%	2.072	2.258	16.800
10	504502	23.30%	2.612	2.650	15.333
11	504524	18.81%	1.990	2.059	16.105
N/A	Baseline	56.45%	3.097	3.556	17.543

Table 2: Overall results of the human evaluation. Bold indicates the best score for each metric.

Experiments

Training Details

We developed our model using the open-source implementation of Wolf et al. (2018)[3] and the `GPT2-small` (124M parameters) that consists of 12 transformer decoder blocks and pre-trained weights (Wolf et al. 2019)[4]. We tokenized each sentence into sub-word using the `GPT2Tokenizer`[4] (Sennrich, Haddow, and Birch 2016).

We fine-tuned the pre-trained GPT-2 model with batch size 2 for 4 epochs over the MultiWOZ training dataset. The maximum history size of each dialogue is set to 15. The optimizer was Adam (Kingma and Ba 2015) with learning late of 6.25e-5, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The coefficients of language modeling loss and next-utterance classification loss were set to 2.0 and 1.0, respectively.

Evaluation Metrics

There are two criteria for evaluating dialogue systems in the first sub-task of Track 1 in DSTC8:

- Automatic evaluation: Success Rate, Return, Turns, Precision, Recall, F1, Book Rate
- Human evaluation: Success Rate, Language Understanding Score, Response Appropriateness Score, Turns

For evaluating the success rate, the dialogue is considered as a success only if all *informable* slots and *requestable* slots are correctly filled. Return is a reward signal obtained from the user simulator when the dialogue is complete. The return of each dialogue is computed as follows:

$$\text{Return} = -\text{Turns} + \begin{cases} 2 * \text{max\_turn} & \text{If task success,} \\ (-1) * \text{max\_turn} & \text{otherwise.} \end{cases}$$

The max_turn means the maximum limit of turns in conversation (e.g. 40). Precision, Recall, and F1 consider how requestable slots are filled. Language Understanding Score and Response Appropriateness Score are the metrics of how natural the response of the model is, with the 5 point scale.

Both evaluations are ranked by success rate and the final ranking is determined through the human evaluation. The table 1 and 2 shows the overall results of automatic and human evaluation in ConvLab official website.[5] The dialogue

[3]https://github.com/huggingface/transfer-learning-conv-ai

[4]https://github.com/huggingface/transformers

[5]https://convlab.github.io/

system provided as the baseline is a rule-based pipelined dialogue system, in which only the NLU part is replaced with MILU.

Automatic Evaluation

Our proposed model with top-p sampling (p=0.8) ranked 5th place with the success rate of 79.40%, the return of 49.69, the turns of 7.59, the book rate of 87.02%, the precision of 0.8, the recall of 0.89, and the F1 score of 0.83 at the automatic evaluation phase. The automatic evaluation was only used to filter out low-performing submissions because the user simulator was inherently not perfect.

Human Evaluation

As the final ranking, our proposed model with top-p sampling (p=0.8) ranked 1st place with the success rate of 68.32%, the turn of 19.507, the language understanding score of 4.149 and the response appropriateness score 4.287. Compared to the 2nd-ranked model, Our model showed a 2.51% improvement in success rate. Especially our model outperforms on the human language metrics, 0.365 points higher than the 2nd-ranked model in the Language Understanding score, and 0.447 points higher than the 2nd-ranked model in the Response Appropriateness score.

Conclusion

In this paper, we presented an end-to-end monolithic neural model for goal-oriented dialogues that simulates the core steps of the dialogue management pipeline. Our work can be adopted for any large pre-trained models such as BERT (Devlin et al. 2019) and XLNet (Yang et al. 2019). Since our model outputs all the intermediate results in the dialogue management pipeline, it is easy to integrate with external systems as well as to identify the point of failure for unreasonable responses. The experimental results with human evaluation demonstrate that our model can provide very natural human-level interaction for goal-oriented dialogues, advancing the state-of-the-art in conversational AI agents.

Acknowledgements

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant (19ZS1100), the National Research Foundation (NRF) of Korea (NRF-2019R1A2C1087634), the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program (No. 10063424), and Samsung Research.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A Neural Probabilistic Language Model. *Journal of machine learning research*.

Budzianowski, P., and Vulić, I. 2019. Hello, It's GPT-2 – How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*.

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gasic, M. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Chen, W.; Chen, J.; Qin, P.; Yan, X.; and Wang, W. Y. 2019. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Dinan, E.; Logacheva, V.; Malykh, V.; Miller, A.; Shuster, K.; Urbanek, J.; Kiela, D.; Szlam, A.; Serban, I.; Lowe, R.; et al. 2018. The Second Conversational Intelligence Challenge (ConvAI2). In *The NeurIPS'18 Competition*.

Holtzman, A.; Buys, J.; Forbes, M.; and Choi, Y. 2019. The Curious Case of Neural Text Degeneration. *arXiv preprint* abs:1904.09751.

Kim, Y.-B.; Lee, S.; and Stratos, K. 2017. OneNet: Joint Domain, Intent, Slot Prediction for Spoken Language Understanding. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

Kingma, D. P., and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

Lee, S.; Zhu, Q.; Takanobu, R.; Li, X.; Zhang, Y.; Zhang, Z.; Li, J.; Peng, B.; Li, X.; Huang, M.; and Gao, J. 2019. ConvLab: Multi-Domain End-to-End Dialog System Platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

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*.

Lei, W.; Jin, X.; Kan, M.-Y.; Ren, Z.; He, X.; and Yin, D. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Madotto, A.; Wu, C.-S.; and Fung, P. 2018. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Mehri, S.; Srinivasan, T.; and Eskenazi, M. 2019. Structured fusion networks for dialog. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Pei, J.; Ren, P.; and de Rijke, M. 2019. A modular task-oriented dialogue system using a neural mixture-of-experts.

In WCIS: SIGIR 2019 Workshop on Conversational Interaction Systems.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. https://www.techbooky.com/wp-content/uploads/2019/02/Better-Language-Models-and-Their-Implications.pdf.

Ramadan, O.; Budzianowski, P.; and Gašić, M. 2018. Large-Scale Multi-Domain Belief Tracking with Knowledge Sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Wen, T.-H.; Gašić, M.; Mrkšić, N.; Su, P.-H.; Vandyke, D.; and Young, S. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Wen, T.-H.; Vandyke, D.; Mrkšić, N.; Gašić, M.; Rojas-Barahona, L. M.; Su, P.-H.; Ultes, S.; and Young, S. 2017. A Network-based End-to-End Trainable Task-oriented Dialogue System. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Weston, J.; Dinan, E.; and Miller, A. 2018. Retrieve and Refine: Improved Sequence Generation Models For Dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*.

Williams, J.; Raux, A.; Ramachandran, D.; and Black, A. 2013. The Dialog State Tracking Challenge. In *Proceedings of the SIGDIAL 2013 Conference*.

Wolf, T.; Sanh, V.; Chaumond, J.; and Delangue, C. 2018. TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents. *NeurIPS 2018 workshop on Conversational AI: "Today's Practice and Tomorrow's Potential"*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; and Brew, J. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint* abs:1910.03771.

Wu, C.-S.; Madotto, A.; Hosseini-Asl, E.; Xiong, C.; Socher, R.; and Fung, P. 2019. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*.

Zhao, T.; Xie, K.; and Eskenazi, M. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.