# End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems 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 goal-oriented dialogue system needs to be optimized for tracking the dialogue flow and carrying out an effective conversation under various situations to meet the user goal. The traditional approach to building such a dialogue system is to take a pipelined modular architecture, where its modules are optimized individually. However, such an optimization scheme does not necessarily yield an overall performance improvement of the whole system. On the other hand, end-to-end dialogue systems with monolithic neural architecture are often 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 be integrated with external systems or to provide interpretable information about why the system generated a particular response. In this paper, we present an end-to-end neural architecture for dialogue systems that addresses both challenges above. 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 in human evaluations, which ranked the system at the top position in the end-to-end multi-domain dialogue system task in the 8th dialogue systems technology challenge (DSTC8).

## 1 Introduction

The goal-oriented dialogue system helps users achieve their goals such as requesting information or executing commands via natural language conversations. It is thus 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 the dialogue flow is suddenly changed.

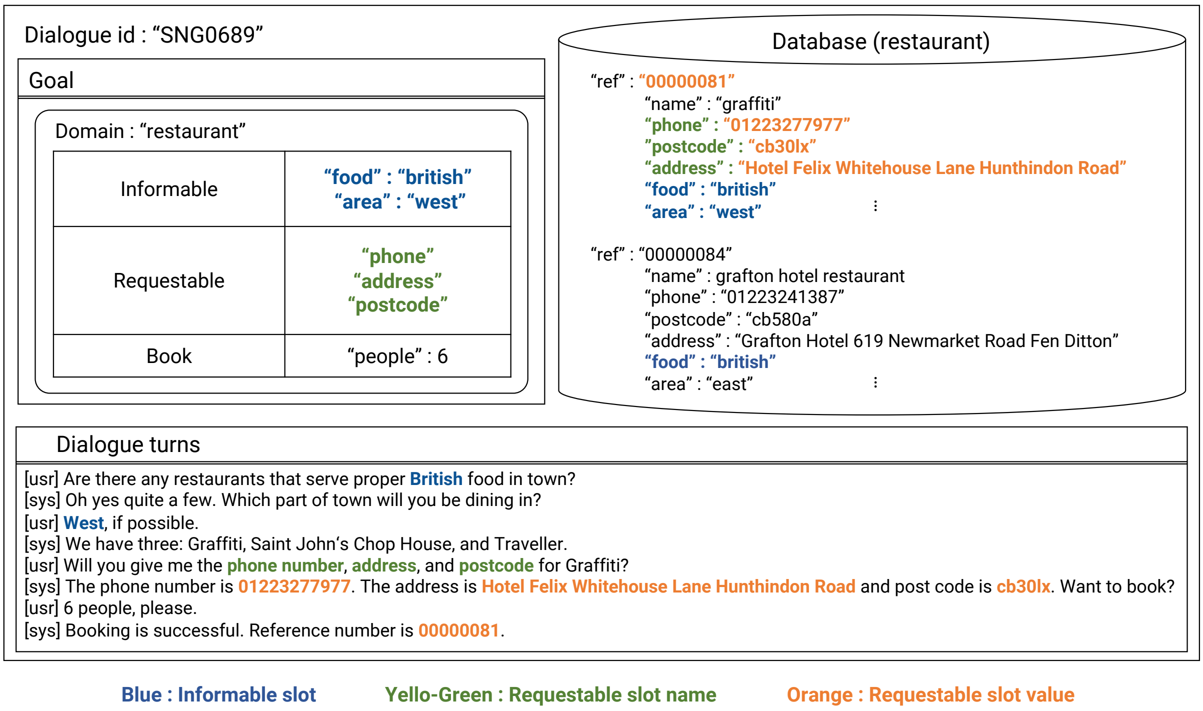
The traditional approach to building a goal-oriented dialogue system mostly adopts a pipelined modular architecture, with the natural language understanding (NLU) module (Kim et al., 2017; Lee et al., 2019b) that first recognizes and comprehends user's intent and extracts values for slots, then the dialogue 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 et al., 2018; Wu et al., 2019; Lee et al., 2019a) which maps the dialogue history to the dialogue state (the composite function of NLU and DST), and the Word-level POL (Budzianowski et al., 2018; Pei et al., 2019; Chen et al., 2019; Mehri et al., 2019; Zhao et al., 2019) which maps the previous utterance and dialogue state to the system response (the composite function of POL and NLG).

These modules are usually optimized separately, which does not necessarily lead to an overall optimized performance for successful task completion. On the other hand, end-to-end neural models for dialogue systems (Madotto et al., 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.

In this paper, we present an end-to-end neural architecture for dialogue systems that addresses both challenges above. Our work is based on fine-tuning GPT-2 (Radford et al., 2019) to faithfully perform the following essential dialogue management steps in a sequential manner using a *single* model: (1)

[*] : Equal contribution

*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592
July 5 - 10, 2020. ©2020 Association for Computational Linguistics



| | |
|---|---|
| Informable | "food" : "british"<br>"area" : "west" |
| Requestable | "phone"<br>"address"<br>"postcode" |
| Book | "people" : 6 |

"ref" : "00000081"
"name" : "graffiti"
"phone" : "01223277977"
"postcode" : "cb30lx"
"address" : "Hotel Felix Whitehouse Lane Hunthindon Road"
"food" : "british"
"area" : "west"
⋮

"ref" : "00000084"
"name" : grafton hotel restaurant
"phone" : "01223241387"
"postcode" : "cb580a"
"address" : "Grafton Hotel 619 Newmarket Road Fen Ditton"
"food" : "british"
"area" : "east"
⋮

[usr] Are there any restaurants that serve proper British food in town?
[sys] Oh yes quite a few. Which part of town will you be dining in?
[usr] West, if possible.
[sys] We have three: Graffiti, Saint John's Chop House, and Traveller.
[usr] Will you give me the phone number, address, and postcode for Graffiti?
[sys] The phone number is 01223277977. The address is Hotel Felix Whitehouse Lane Hunthindon Road and post code is cb30lx. Want to book?
[usr] 6 people, please.
[sys] Booking is successful. Reference number is 00000081.

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

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 system action, and (4) NLG via predicting the system response. As a result, our neural model not only generates the system response just like end-to-end neural dialogue systems, but also generates dialogue states and system actions as intermediate outputs, improving the interpretability of the behavior of the dialogue system. In order to achieve this, we leverage the annotations of dialogue states and system actions provided in the corpus (e.g. MultiWOZ dataset (Budzianowski et al., 2018)) for training our system in a very natural way.

Our model is evaluated using ConvLab (Lee et al., 2019b), a multi-domain end-to-end dialog system platform to support various aspects in the development and evaluation of dialogue systems, in terms of the automatic evaluation using the user simulator and the human evaluation using crowd workers. Particularly, in the human evaluation carried out as a part of the 8th dialogue systems technology challenge (DSTC8) (Kim et al., 2019), our system attained the success rate of 68.32%, the language understanding score of 4.149, and the response appropriateness score of 4.287, ranking at the 1st place in DSTC8. We also show that our model is competitive to other state-of-the-art models specialized for two sub-tasks in the dialogue management, i.e. Dialogue State Tracking and Dialogue-Context-to-Text Generation tasks, although our model was not particularly tuned for those sub-tasks.

The main characteristics of our model can be summarized as follows: (1) it is trained to follow the traditional dialogue management pipeline, making the monolithic neural model more interpretable and easily integratable with external systems, while (2) it is trained in an end-to-end fashion with simple gradient descent, and (3) leverages GPT-2, a powerful pre-trained language model. The code is available through the GitHub code repository.[1]

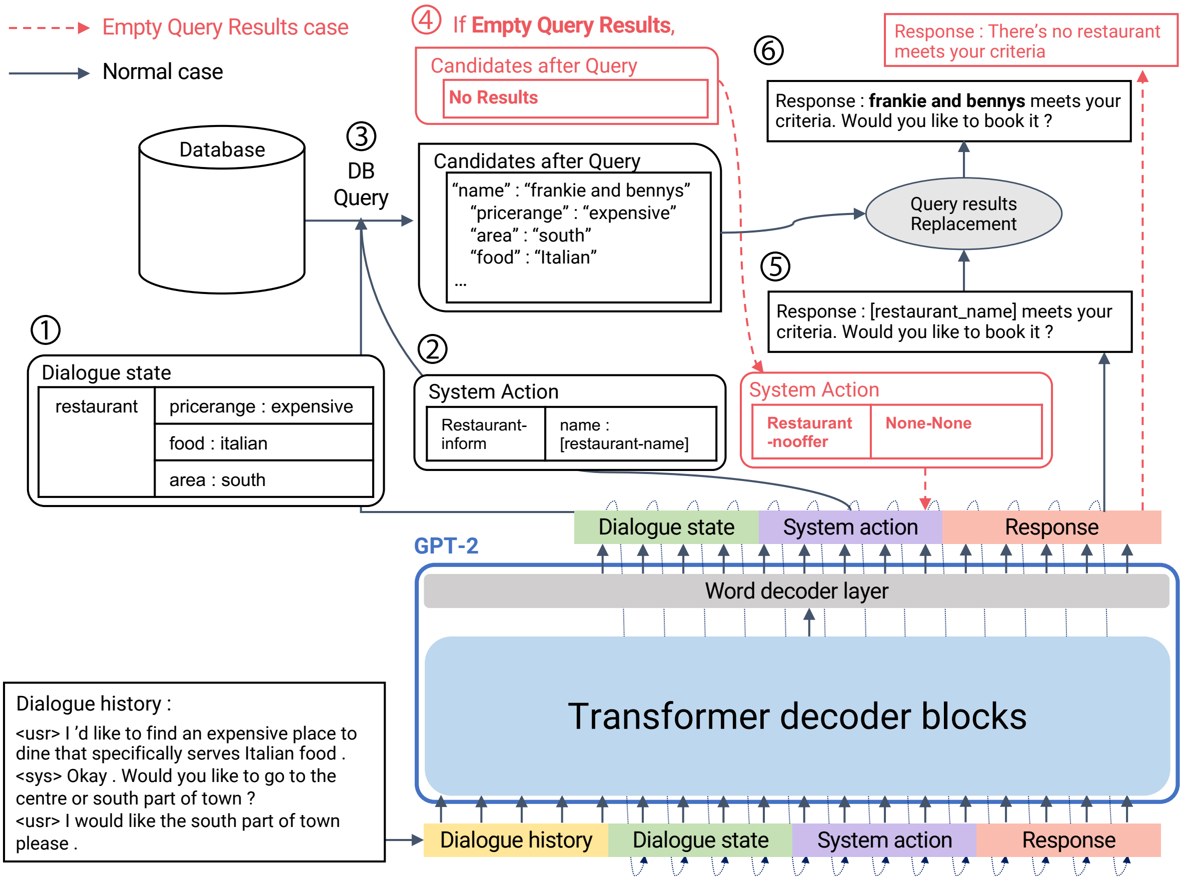
## 2 End-to-end Multi-Domain Task-Completion Task

Before we describe our approach, we briefly overview the end-to-end multi-domain task-completion task used in DSTC8, for which we developed our dialogue system.

### 2.1 The MultiWOZ Dataset

The MultiWOZ dataset is a large-scale fully annotated corpus of natural human-human conversa-

[1] https://github.com/KAIST-AILab/NeuralPipeline_DSTC8



Empty Query Results case
Normal case
④ If Empty Query Results,
Candidates after Query
No Results
Response : There's no restaurant meets your criteria
⑥
Response : frankie and bennys meets your criteria. Would you like to book it ?
Database
③ DB Query
Candidates after Query
"name" : "frankie and bennys"
"pricerange" : "expensive"
"area" : "south"
"food" : "Italian"
...
Query results Replacement
⑤
Response : [restaurant_name] meets your criteria. Would you like to book it ?
①
Dialogue state
restaurant
pricerange : expensive
food : italian
area : south
②
System Action
Restaurant-inform
name : [restaurant-name]
System Action
Restaurant -nooffer
None-None
Dialogue state
System action
Response
GPT-2
Word decoder layer
Transformer decoder blocks
Dialogue history :
<usr> I 'd like to find an expensive place to dine that specifically serves Italian food .
<sys> Okay . Would you like to go to the centre or south part of town ?
<usr> I would like the south part of town please .
Dialogue history
Dialogue state
System action
Response

Figure 2: The overview of our end-to-end neural dialogue model. For the transformer, we use fine-tuned 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.

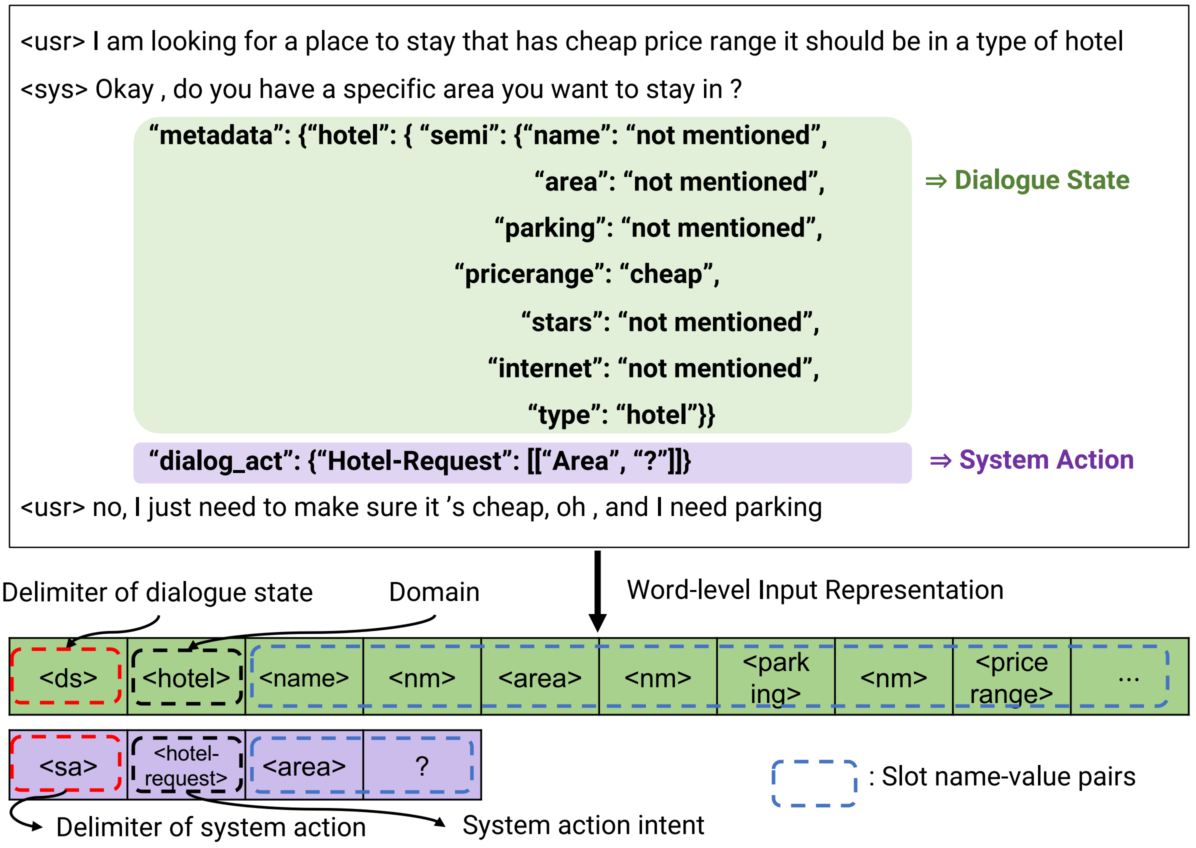
tions, where the user as a tourist converses with the system as a clerk across multiple domains. Each dialogue is rich in annotations such as 'goal', 'metadata', and 'dialog act' as well as user and system utterances. These annotations facilitate using machine learning to develop individual modules of a dialogue system (NLU, DST, POL, NLG, Word-level DST, Word-level POL), as well as an end-to-end dialogue system.

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 domain and the slots. The slots are divided into *informable*, *requestable* and *book* slots. *Informable* slots represent user constraints and *Requestable* slots hold additional information that the user wants to obtain. *Book* slots are used to reserve a place recommended by the system.

## 2.2 ConvLab

For evaluating dialogue systems, DSTC8 used ConvLab (Lee et al., 2019b), an open-source platform that supports researchers to train and evaluate their own dialogue systems. ConvLab contains implementations of the state-of-the-art models of NLU, DST, POL, NLG (Kim et al., 2017; Lee et al., 2019b; Ramadan et al., 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 et al., 2018), which are readily reusable for building dialogue systems using various approaches.

ConvLab also provides an agenda-based user simulator to easily interact with the target dialogue system, consisting of a multi-intent language understanding(MILU) (Lee et al., 2019b) for NLU, a rule-based policy, and a template-based NLG. For each dialogue, a goal is randomly 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



<usr> I am looking for a place to stay that has cheap price range it should be in a type of hotel

<sys> Okay , do you have a specific area you want to stay in ?

“metadata”: {“hotel”: { “semi”: {“name”: “not mentioned”,
“area”: “not mentioned”,
“parking”: “not mentioned”,
“pricerange”: “cheap”,
“stars”: “not mentioned”,
“internet”: “not mentioned”,
“type”: “hotel”}}

“dialog_act”: {“Hotel-Request”: [[“Area”, “?”]]}

<usr> no, I just need to make sure it ’s cheap, oh , and I need parking

Figure 3: In the MultiWOZ dataset, the ‘metadata’ is treated as the dialogue state and the ‘dialogue act’ is treated as the system action.

the target dialogue system, it recognizes the system dialogue act, decides the user dialogue act from the agenda stack, and generates the user response at each turn. When the system offers to book and the user accepts it, the system should notify an 8-digit reference number. The reference number is used to verify whether the booked place is fit on what the user informs. ConvLab also provides an automatic evaluator which assesses whether the target dialogue system (1) traces what the user informs (2) informs what the user requests, and (3) makes an appropriate booking using an external database based on the traced information. Although the user simulator and evaluator are highly sophisticated, it is not as perfect as human. Hence, the dialogue systems submitted to the DSTC8 were evaluated not only with the user simulator but also with human crowd-workers.

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

We now describe our end-to-end neural pipeline for the goal-oriented dialogue system based on GPT-2. Our system consists of (1) the GPT-2 model fine-tuned on the delexicalized version of MultiWOZ dataset (Section 3.2) and (2) the database query module. We take the pre-trained GPT-2 model and fine-tune it to follow the steps of the dialogue management pipeline. Figure 2 illustrates an overall architecture with a concrete example. The overview of the process followed by our model is as follows:

1. Predict the recent domain and the corresponding dialogue state conditioned on the dialogue history.
2. Predict the system action with delexicalized tokens conditioned on the dialogue history and dialogue state.
3. If the 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 when detecting Empty Query Results (Section 3.5).
5. Generate the system response with delexicalized tokens conditioned on dialogue history,

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

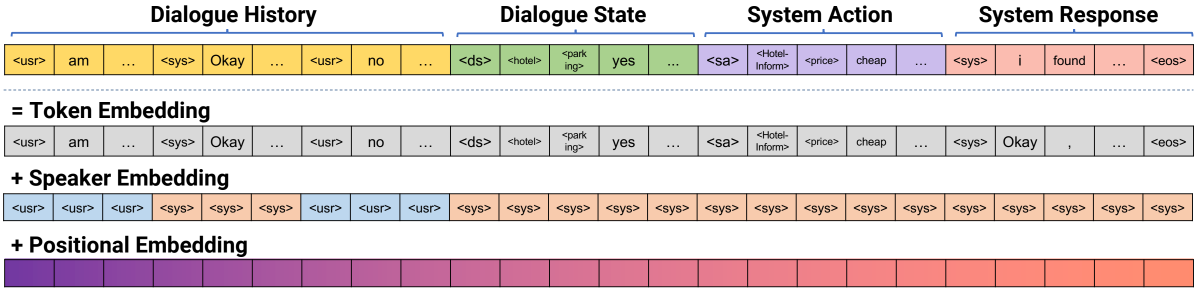

Dialogue History
Dialogue State
System Action
System Response
<usr> am … <sys> Okay … <usr> no … <ds> <hotel> <parking> yes … <sa> <Hotel-Inform> <price> cheap … <sys> i found … <eos>
= Token Embedding
<usr> am … <sys> Okay … <usr> no … <ds> <hotel> <parking> yes … <sa> <Hotel-Inform> <price> cheap … <sys> Okay , … <eos>
+ Speaker Embedding
<usr> <usr> <usr> <sys> <sys> <sys> <usr> <usr> <usr> <sys> <sys> <sys> <sys> <sys> <sys> <sys> <sys> <sys> <sys> <sys> <sys> <sys> <sys> <sys>
+ Positional Embedding

Figure 4: Input representation for fine-tuning GPT-2.

dialogue state, and system action.

6. Update the delexicalized tokens in the system response with the query result.

In Figure 2, the numbers wrapped with circle indicate the order of process. The red box shows how our system handles the case when the DB query does not return any record at all.

### 3.1 Input Representation

In the MultiWOZ dataset, 'metadata' and 'dialog_act' correspond to the current dialogue state and the current system action, respectively (Figure 3). In order to use GPT-2, we need to convert the dialogue state and the system action to word tokens.

Figure 3 shows an illustrative example of a single-turn of a dialogue and its representation of the dialogue state and system action. We introduce delimiter tokens <usr>, <sys>, <ds> and <sa> to signal the beginning of sequence representations of user utterance, system response, dialogue state, and system action. The domain and the slot names are also represented by additional special tokens, and <nm> and <dc> are special tokens that indicate '*not mentioned*' and '*don't care*'.

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

### 3.2 Delexicalization

Each dialogue in MultiWOZ dataset is generated based on the DB query results, and as such, the requestable slot values such as reference numbers and addresses (e.g. those colored in orange in Figure 1) are valid only for that particular dialogue instance. On the other hand, our model should be able to inform appropriate information depending on the dialogue context. To address this, we delexicalized all the values for requestable slots (reference number, name, postcode, phone number, address) as [DOMAIN_SLOTNAME] (e.g. [hotel_postcode] for hotel's postcode) that appear in the corpus. Thus, our model learns to generate delexicalized system response, and delexicalized tokens are later string-replaced by the real information from the DB query using a small piece of post-processing code.

### 3.3 Training Objective

In order to fine-tune GPT-2, we optimize the weighted sum of the objectives of language modeling (LM) and next-utterance classification (NC), following (Radford et al., 2018). For 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 calculates the likelihood of the next word-token from given the previous word-tokens.

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

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

| Model | Success Rate ↑ | Return ↑ | Turns ↓ | Precision ↑ | Recall ↑ | F1 ↑ | Book Rate ↑ |
|---|---|---|---|---|---|---|---|
| Baseline | 62.00% | 28.22 | 8.18 | 0.70 | 0.83 | 0.74 | 84.38% |
| **Ours + greedy** | **78.60%** | **48.92** | 7.40 | 0.87 | **0.89** | **0.87** | **86.34%** |
| Ours + top-p (p=0.8) | 75.40% | 44.67 | 7.81 | **0.88** | 0.88 | 0.86 | 84.10% |
| Ours + top-k (k=30) | 74.80% | 44.47 | **7.29** | 0.83 | 0.86 | 0.83 | 83.49% |

Table 1: Results of decoding strategies in the automatic evaluation, using the ConvLab evaluator. A baseline system provided by ConvLab consists of MILU (Lee et al., 2019b) as NLU module, rule-based DST and POL, and template-based NLG.

| 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 |
| N/A | Baseline | 56.45% | 3.097 | 3.556 | 17.543 |

Table 2: Overall results of the human evaluation carried out by DSTC8 organizers. Only the top five teams and the baseline results are compared.

### 3.4 Decoding Strategy

When we generate the system response from the dialogue history, the final output is the probability distribution of word-tokens at each position. Using the distribution, there are many decoding methods for generating word-tokens, which have a significant impact on the quality of the output (Holtzman et al., 2020; Weston et al., 2018). The greedy decoding and the beam search are the most common approaches. However, since the greedy decoding only considers the token with the highest probability at each position, it does not necessary yield a system response with overall high probability. In addition, Holtzman et al. (2020) 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* have been shown to 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 the decoding schemes mentioned above, and selected the best one via human evaluation.

### 3.5 Handling Empty Query Result

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 happens 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.

## 4 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 the 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 towards DSTC8 submission, Budzianowski and Vulić (2019) also demonstrated a neural model for goal-oriented dialogue systems by fine-tuning GPT-2 on the MultiWOZ dataset. However, they only handle dialogue-context-to-text task, which outputs the system response given the dialogue history, the *ground-truth* dialogue state, and the database. In our case, no oracle information related to database

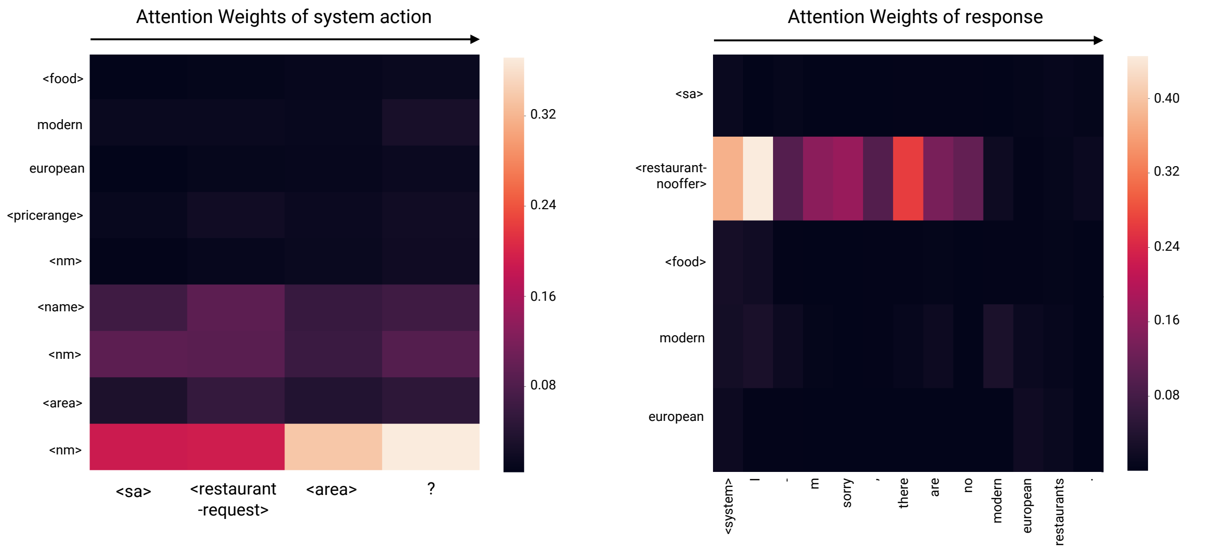


Figure 5: Visualizing attention weights. (*left*) The model attends to the dialogue state `<area> <nm>` for generating system action `<restaurant-request> <area>`. (*right*) The model attends to the system action `<restaurant-nooffer>` for generating response '*I'm sorry. There are no modern European restaurants*'.

and dialogue state is provided, and only the dialogue history was provided. Taking the dialogue history as an input, our model operates as a complete dialogue system that generates system responses by sequentially following the core steps in the dialogue management pipeline.

# 5 Experimental Settings

## 5.1 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 `GPT2Tokenizer`[4] (Sennrich et al., 2016).

We fine-tuned the GPT-2 with batch size 2 for 4 epochs over the MultiWOZ training dataset. The maximum history size of each dialogue was set to 15. We used the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and the learning late of 6.25e-5. The coefficients of the LM and the NC losses were set to 2.0 and 1.0, respectively.

## 5.2 Evaluation Metrics

There were two evaluation criteria in the End-to-End Multi-Domain Dialog System Task of the Multi-Domain Task-Completion Track in DSTC8:

- Automatic evaluation with user simulator: Success Rate, Book Rate, Return, Turns, Precision, Recall, F1
- Human evaluation with crowd-workers: Success Rate, Language Understanding Score, Response Appropriateness Score, Turns

In measuring the success rate, the dialogue is considered as a success only if the *requestable* slots are correctly filled and *book success* if needed. *Book success* is achieved only if the reserved information fits into all *informable* slots, and is measured by the book rate as a sub-evaluation.

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 indicates the maximum limit of turns in a conversation (e.g. 40). Precision, Recall, and F1 measure the accuracy of *requestable* slot filling.

For the human evaluation, Language Understanding Score and Response Appropriateness Score were the metrics of how natural the response of the model is, with the 5 point scale. The human evaluation results reported here were carried out by the DSTC8 organizers.

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

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

# 6 Results

## 6.1 Automatic Evaluation

Table 1 shows automatic evaluation results on various decoding strategies using the user simulator provided in ConvLab. Our proposed model with greedy decoding strategy achieved the success rate of 78.60%, the avg return of 48.92, the avg turns of 7.40, the book rate of 86.34%, the precision of 0.87, the recall of 0.89, and the F1 score of 0.87 in the automatic evaluation using 500 simulated dialogues. Our model outperformed the baseline system, but failed to perform best among submitted systems, mostly due to the incorrect intent recognition in the user simulator. We believe that this can be circumvented by further training our model using reinforcement learning, trained to avoid system responses that trigger intent recognition failure in the simulator. However, our main focus was to generate diverse system responses that looked natural to human evaluators.

## 6.2 Human Evaluation

Table 2 shows the final ranking of the competition using human evaluation.[5] Our proposed model with top-p sampling (p=0.8) strategy ranked in the first place with the success rate of 68.32%, the average turns 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. The performance gap was more significant in human language metrics, 0.365 points and 0.458 points higher than the 2nd-ranked model in the Language Understanding score and the Response Appropriateness score.

## 6.3 Attention Weights

Figure 5 visualizes the attention weights of the transformer blocks in our model, demonstrating that our model appropriately attends to the word token generated from the previous module in the dialogue management pipeline, just like a pipelined dialogue system would do when generating the intermediate outputs. For example, if the user asks '*I'm looking for modern European food*', our model generates dialogue state `<area>` `<nm>`, which means the area is not mentioned. Then we can see the attention weight on `<area>` `<nm>` in the dialogue state is relatively higher than other tokens when it generates system action `<restaurant-request>` `<area>`. As another example, if we change the system action as `<restaurant-nooffer>`, the model generates the system response '*I'm sorry. There are no modern European restaurant*' and it attends on the token `<restaurant-nooffer>`.

| Model | Joint Acc. | Slot Acc. |
|---|---|---|
| GLAD<br>(Zhong et al., 2018) | 35.57 | 95.44 |
| GCE<br>(Nouri and Hosseini-Asl, 2018) | 36.27 | 98.42 |
| SUMBT<br>(Lee et al., 2019a) | 46.64 | 96.44 |
| TRADE<br>(Wu et al., 2019) | **48.62** | **96.92** |
| **Ours + greedy** | 44.03 | 96.07 |

Table 3: Performance comparison with other state-of-the-art models in Dialogue State Tracking benchmark of MultiWOZ dataset.

| Model | Inform | Success | BLEU |
|---|---|---|---|
| Baseline<br>(Budzianowski et al., 2018) | 71.29 | 60.96 | 18.80 |
| TokenMoE<br>(Pei et al., 2019) | 75.30 | 59.70 | 16.81 |
| HDSA<br>(Chen et al., 2019) | 82.9 | 68.90 | **23.60** |
| Structured Fusion<br>(Mehri et al., 2019) | 82.70 | 72.10 | 16.34 |
| LaRL<br>(Zhao et al., 2019) | **82.78** | **79.20** | 12.80 |
| **Ours + greedy** | 77.00 | 69.20 | 6.01 |

Table 4: Performance comparison with other state-of-the-art models in Dialogue-Context-to-Text Generation benchmark of MultiWOZ dataset.

## 6.4 MultiWOZ Benchmarks Performance

As an ablation study, we test the modular performance of our model on two MultiWOZ benchmark tasks (Budzianowski et al., 2018): Dialogue State Tracking and Dialogue-Context-to-Text Generation.

### 6.4.1 Dialogue State Tracking

Table 3 compares the dialogue state tracking accuracy of our model to those of other recent trackers in the literature. In this task, we measure the joint accuracy and slot accuracy of dialogue state tracking part of our model. Although our training objective involves other dialogue management tasks than dialogue state tracking, our model's tracking perfor-

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

mance was very competitive to the state-of-the-art models.

### 6.4.2 Dialogue-Context-to-Text Generation

Dialogue-Context-to-Text generation looks at the combined performance of the dialogue policy and the system response generation modules, measuring the quality of system response when the previous user utterance, the *ground-truth* dialogue state, and the *ground-truth* database query results are given. Our trained model can be straightforwardly adapted to perform this task by replacing the intermediate inputs with ground-truth values.

Table 4 shows the Context-to-Text Generation benchmark performance compared to other recent models proposed in the literature. Again, our model was competitive to the state-of-the-art models except for the BLEU score. This is due to the fact that the system uses the large vocabulary of GPT-2, making system responses often containing diverse words that are not in the dataset.

## 7 Conclusion

In this paper, we presented an end-to-end monolithic neural model for goal-oriented dialogues that learns to follow the core steps in the dialogue management pipeline. Since our model outputs all the intermediate results in the dialogue management pipeline, it is easy to integrate with external systems and to interpret why the system generates a particular response. The experimental results from human evaluation show evidence that our approach can provide very natural human-level interaction for goal-oriented dialogues, advancing the state-of-the-art in conversational AI agents. This also demonstrates the power of large-scale pre-trained language models to be adopted for building end-to-end goal-oriented dialogue systems.

## Acknowledgements

This work was supported by the National Research Foundation (NRF) of Korea (NRF-2019R1A2C1087634) and the Ministry of Science and Information communication Technology (MSIT) of Korea (IITP No. 2020-0-00940, IITP 2019-0-00075-001 and IITP No. 2017-0-01779 XAI).

## References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*.

Paweł Budzianowski and Ivan Vulić. 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*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2018. The Second Conversational Intelligence Challenge (ConvAI2). In *The NeurIPS'18 Competition*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, Minlie Huang, Luis Lastras, Jonathan K. Kummerfeld, Walter S. Lasecki, Chiori Hori, Anoop Cherian, Tim K. Marks, Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, and Raghav Gupta. 2019. The eighth dialog system technology challenge. In *Third NeurIPS workshop on Conversational AI: "Today's Practice and Tomorrow's Potential"*.

Young-Bum Kim, Sungjin Lee, and Karl Stratos. 2017. OneNet: Joint Domain, Intent, Slot Prediction for Spoken Language Understanding. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019a. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Xiang Li, Yaoqin Zhang, Zheng Zhang, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019b. ConvLab: Multi-Domain End-to-End Dialog System Platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 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*.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking. In *NeurIPS 2018, 2nd Conversational AI workshop*.

Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. 2019. A modular task-oriented dialogue system using a neural mixture-of-experts. *In WCIS: SIGIR 2019 Workshop on Conversational Interaction Systems*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 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*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *https://www.techbooky.com/wp-content/uploads/2019/02/Better-Language-Models-and-Their-Implications.pdf*.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 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*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Jason Weston, Emily Dinan, and Alexander Miller. 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*, Brussels, Belgium. Association for Computational Linguistics.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The Dialog State Tracking Challenge. In *Proceedings of the SIGDIAL 2013 Conference*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint abs:1910.03771*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 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"*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 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*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.