ADL Final Report
Team 22

Abstract

Task-oriented dialogue system consists of three main subtasks: dialogue state tracking (DST), dialogue management and response generation. In this report, we focus on two tasks, DST and generating chit-chat responses aside from the original response generation. For the dialogue state tracking task, we propose two generative models to solve the problem on both seen and unseen domains, additionally, we discuss the effect of utilizing the information of the slot of domain. For the chit-chat generation, we train our model on the given chit-chat dataset, and allow it to generate human-like chit-chat responses by a decent frequency.

1 Introduction

The advancement of human-computer interaction relies on the development of the algorithm of artificial intelligence. How to understand people's needs and give a reasonable response is a challenging task for a task-oriented dialogue system. Within the three subtasks of the task-oriented dialogue system, DST is the first step which interprets human language and determines the current belief state based on the user's request or information. Due to the increasing variety of users' needs, the systems have to support new domains. However, collecting and annotating training data for DST tasks is relatively expensive and time-consuming, therefore, it is important for the model to deal with problems in unseen domains, given some data in currently seen domains.

Moreover, as the systems achieve a certain quality on generating task-related responses, they can be improved in other aspects, such as making responses richer and more interesting. The difference between humans and task-oriented bots is that bots only focus on content related to the task, but humans share their emotions, propose their own ideas, or provide additional information. Thus, one way to create a more human-like response is to add these extra contents, so called "chit-chat", to the existing task-oriented response. Another task then appeared: which position and at what frequency should the chit-chat be placed?

In this report, given the dialogue dataset with chit-chat from [1], we proposed two models, T5DST and a GPT2-based model, trained for DST task, which also tackle dialogues in unseen domains; we trained another GPT2-based model, which generate chit-chat for a given dialogue turn, and invent a method to determine chit-chat position and appearing frequency.

2 Approach

2.1 DST Task

For the DST Task, within several models we tried, there are two approaches that we finally succeeded with, which are T5DST and GP2-based model.

2.1.1 T5DST

The design of T5DST refers to [2], which follows the basis of sequence to sequence question-answering model.

2.1.1.1 Preprocessing

For the input, given a dialogue history consisting of alternating set of utterances from two speakers, user and system, denoted as $D_t = \{U_1, S_1, ..., S_{t-1}, U_t\}$, and a slot of domain, denoted as s_i of d, we add a prefix "user:" to the user utterance and a prefix "system:" to the system utterance. Then we concatenate the dialogue history, and append a separate token, the slot of domain, and the description of the slot at the end, i.e., "user: U_1 system: S_1 ...system: S_{t-1} user: U_t [SEP] s_i of d:[description]". The truth slot value $\widehat{v_i}$ followed by an end-of-sentence token [EOS] is the expected output, the purpose of the end-of-sentence token is to prevent the model from generating length exceeding outputs.

2.1.1.2 Model

$$v_i = T5DST(D_t,\ s_i,\ d,\ description_{s_i})$$

For the model, we use T5ForConditionalGeneration from huggingface, with pretrained parameters T5-small, which has 60M parameters, 6 encoder-decoder layers, and hidden size 512.

2.1.1.3 Training and Generation

The model is trained with initial learning rate 0.0001 and batch size 8 for 5 epochs. For the generation process, we use beam search with beam size 3.

2.1.2 GPT2

The design of our GPT2-based model refers to SimpleTOD [3] which is a single end-to-end model that deals with three subtasks in task-oriented dialogue: dialogue state tracking, dialogue management and response generation. We utilize the part for dialogue state tracking in our model.

2.1.2.1 Preprocessing

Given a dialogue history consisting of alternating set of utterances from two speakers, user and system, denoted as $D_t = \{U_1, S_1, ..., S_{t-1}, U_t\}$, and slot of domain, denoted as s_i of d. For a uncategorical slot of domain s_i of d, we first construct a formatted natural language question q_i = "What is the s_i, $description_{s_i}$, of the d, $description_d$ "; on the other hand, for a categorical slot of domain s_i of d, and its k possible values $v'_{i,1}$, ..., $v'_{i,k}$, we construct a question q_i ="Which is the s_i, $description_{s_i}$, of the d, $description_d$, $v'_{i,1}$, ..., $v'_{i,k}$?". For the input, we first add special tokens “<|user|>” and “<|system|>” as a prefixes to the user utterance and the system utterance respectively, enclose context, question, and the truth answer $\widehat{v_i}$ with their corresponding special tokens, and concatenate them all together, i.e., “<|context|><|user|> U_1 <|system|> S_1 ...<|user|> U_t <|endofcontext|><|question|> q_i <|endofquestion|><|answer|> $\widehat{v_i}$ <|endofanswer|>”.

2.1.2.2 Model

$$v_i = GPT2(D_t,\ q(\ s_i,\ d,\ description_{s_i,\ d}))$$

For the model, we use GPT2LMHeadModel from huggingface, with pretrained parameters gpt2-medium, which has 345M parameters, 24 layers, and hidden size 1024.

2.1.1.3 Training and Generation

The model is trained with initial learning rate 5×10^{-5}, batch size 8, and gradient accumulation step size 4 for 2 epochs. For the generation process, we use beam search with beam size 3.

2.2 NLG Task

2.2.1 Preprocessing

Given a dialogue turn with a pair of user utterance and its system utterance response, denoted as $T_t = \{U_t,\ S_t\}$, the begin chit-chat and end chit-chat, denoted as $\widehat{B_t}$ and $\widehat{E_t}$ respectively. For the input, we concatenate all four together with separate token [SEP], i.e., “ U_t [SEP] S_t [SEP] $\widehat{B_t}$ [SEP] $\widehat{E_t}$ ”.

2.2.2 Model

$$B_t = GPT2(U_t,\ S_t\ )$$

$$E_t = GPT2(U_t,\ S_t,\ B_t\ )$$

For the model, we use GPT2 pipeline from huggingface, with pretrained parameters gpt2, which has 117M parameters, 12 layers, and hidden size 768.

2.2.3 Generation

For the generation process, we use beam search with beam size 3, since generating with beam size 5 does not seem to improve the quality of sentences a lot and sacrifices time performance. After feeding “ U_t [SEP] S_t ” into GPT2, since the output from GPT2 may contain several sentences, we extract the first sentence from the output by using Natural Language Toolkit, and use it as the candidate for begin chit-chat, denoted as B_t. Then we concatenate B_t to the end, similarly, after feeding " U_t [SEP] S_t [SEP] B_t ", we extract the first sentence to be the candidate for end chit-chat, denoted as E_t.

2.2.3 Chatty Ratio

We define a chatty ratio $R_c \in [0, 1)$ to control the frequency of chit-chat appearing. When a chit-chat candidate is generated for a certain position, we generate a random number ranging from 0 to 1, if the number is less than R_c, the chit-chat is applied, otherwise the chit-chat at the position remains empty. We found that $R_c = 0.5$ makes the system not too chatty but still active, and we apply it to our final result.

3 Experiments

3.1 DST Task

3.1.1 Performance

Model	public seen	private seen	public unseen	private unseen
GPT2	0.22523	0.22302	0.05992	0.04436
T5DST	**0.30135**	**0.29875**	**0.06741**	**0.08873**

Table 1: Joint goal accuracy for each model on testing datasets

As shown in Table 1, T5DST has a better performance on every dataset for both seen and unseen domains. Both modes have a higher performance on seen domains compared to unseen domains by a larger margin.

3.1.2 Slot Name Format

Format	public seen	private seen	public unseen	private unseen
T5 slot of domain	0.26173	0.28734	0.04494	0.05460
T5 slot of domain + description	**0.30135**	**0.29875**	**0.06741**	**0.08873**
GPT2 SimpleTOD	0.17101	0.15456	0.0000	0.0000
GPT2 Formatted Question	**0.22523**	**0.22302**	**0.05992**	**0.04436**

Table 2: Joint goal accuracy for different slot name format on testing datasets

For T5DST, given a slot of domain, denoted as s_i of d, we've tried two different format representing it, one is purely slot of domain, i.e., "s_i of d", the other includes the description of s_i, i.e., "s_i of d : $description_{s_i}$". As shown in Table 2, including the description improves the performance in all cases.

For the GPT2 base model, we also tried two approaches. The first one follows the original design of SimpleTOD, no slot information is given, the model learned by directly feeding the belief state. The second one is the final approach for GPT2 as we mentioned above, creating a formatted question for the slot of domain. As shown in Table 2, the approach following SimpleTOD totally fails on unseen domains, since the model gets no information about unseen domains and did not learn it before. The second approach, adding a formatted question, not only generates reasonable results on unseen domains, but also outperforms the first approach on seen domains.

3.2 NLG Task

Example 1:
User: Are there any British restaurants in the centre?
System: Oh yes I have quite a few. Did you have a preferred price range in mind? I can recommend some places.

Example 2:
User: Look in Portland.
System: Portland is a great place to visit. How many tickets do you want? Portland is in the pacific northwest, so it should be easy to find.

Example 3:
User: Actually I don't need a reservation at this time.
System: Ok . Is there something else I can help you with then? I'd be happy to answer them for you.

Figure1: Examples of NLG. The sentences marked with orange are the beginning chit-chats, and the ones marked with blue are the end chit-chats.

As shown in Figure 1, there are three examples that we select from our NLG results. In Example 1, the user is asking for a restaurant, and we can see that the chit-chat successfully matches the topic and expresses its willingness to help, which is not task-oriented. In Example 2, the user provides a location for booking tickets, the system first praises the city mentioned by the user at the beginning, then includes additional information about the location, which is correct (Portland is a city located in Oregon which is in the pacific northwest). Although the sentence does not seem to be suitable at this position, this showcases the ability of our model to be knowledgeable about the given user utterance. In Example 3, the user is going to end the service, the system shows its emotion or rather passion to serve the user, which is a more human-like language since we do not often hear words like “happy” or “sad” used by a bot.

The three examples above show that our model is able to generate additional information and human-like sentences while keeping track of the service topic; those behaviors do not appear in an usual task-oriented system.

4 Conclusion

In this report, we propose two generative models, T5DST and a GPT2-based model, for the DST task, tackling dialogues in both seen and unseen domains. We compare the result of using different representations of a slot of domain, and found that including description is an effective method which substantially improves joint goal accuracy. We train another GPT2-based model for chit-chat generation, which can successfully generate chit-chat responses, making the response more human-like and interesting.

5 Work Distribution

DST-T5DST: 陳富中
DST-GPT2: 游一心
NLG: 謝文傑
Report: 鄭達詠

References

[1] Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, Claire Cardie. 2021. Adding Chit-Chat to Enhance Task-Oriented Dialogues. *arXiv preprint arXiv:2010.12757*.

[2] Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu3, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021. Leveraging Slot Descriptions for Zero-Shot Cross-Domain Dialogue State Tracking. *arXiv preprint arXiv:2105.04222*.

[3] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, Richard Socher. A Simple Language Model for Task-Oriented Dialogue. *arXiv preprint arXiv:2005.00796*.