Towards Asking Clarification Questions in Task-Oriented Dialogue

Anonymous ACL submission

Abstract

Task-oriented dialogues aim at providing users with task-specific services. To provide satisfactory services, two major challenges exist: 1) users are not able to fully describe their complex needs due to lack of task knowledge, and; 2) systems need to personalize the service to their users since different users have different profiles and preferences. In order to solve these challenges, systems need to be able to ask questions so as to clarify the user's profile and needs. However, existing task-oriented dialogue systems ignore this aspect. In this paper, we formulate the problem of asking clarification questions in task-oriented dialogue systems. To this end, we propose a dialogue-based user simulator to collect a dataset, called TaskClariQ[1]. We further propose a new System Ask paradigm and a Multi-Attention Seq2Seq Networks (MAS2S) that implements it. Experimental results on TaskClariQ show that MAS2S outperforms competitive baselines.

1 Introduction

While using personal assistant dialogue systems to solve domain-specific tasks, users often fail to formulate their complex request needs. As a consequence, systems may provide inaccurate solutions to users' requests due to the systems inability to know all the needed information about the user request and users themselves (Louvan and Magnini, 2020; Madotto et al., 2020). In other words, a system should always assess its level of confidence for a candidate solution first, and then decide whether to return this solution or ask a clarification question.

Figure 1 shows an example of a task-oriented dialogue. Given a task knowledge, a user profile, and a user request, the task-oriented dialogue system should provide a solution to the user request.

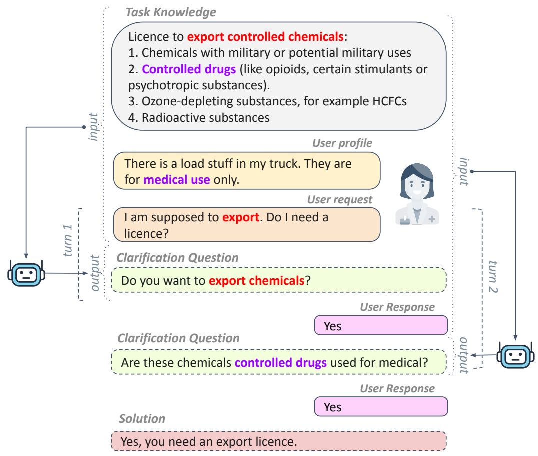

Figure 1: An example of a task-oriented dialogue system asking a clarification question.

In this example we see that the user wants to get help on export. However, in the user request, the user does not mention what goods the user wants to export. Therefore, the system needs to ask a clarification question *"Do you want to export chemicals?"*. From the user profile, the system knows that the goods will be used for medical purposes but does not know whether these goods are controlled drugs. Thus the system needs to ask another clarification question *"Are these chemicals controlled drugs used for medical?"*. The user's responses to the clarification questions aid the system to get a better understanding about the user request. Therefore asking clarification questions based on the task knowledge is crucial in order to provide a more accurate solution for the task-oriented dialogue system.

With the recent advances in neural approaches to conversational AI, researchers have been developing data-driven methods on task-oriented dialogue for either modularized systems or end-to-end systems. For example, RASA (Bocklisch et al., 2017), ConvLab (Zhu et al., 2020), and Conversation Learner (Shukla et al., 2020) are made to allow the use of data-driven approaches based on machine learning to develop dialogue modules. End

[1]To foster research in this area, the dataset and code will be made public upon paper's acceptance.

to-end trainable dialogue systems have also been studied (Budzianowski and Vulić, 2019; Lin et al., 2020; Hosseini-Asl et al., 2020; Yang et al., 2021). Although these methods have achieved promising results, they fail in proactively asking clarification questions to the users in order to clarify user's requests. In regular dialogue systems, clarification question generation solved by generation-based models (Kumar and Black, 2020; Cao et al., 2019) or ranking-based models (Xu et al., 2019; Aliannejadi et al., 2019). However, prior work on clarification question generation ignores task knowledge and task-related user profile, which cannot be directly applied on task-oriented dialogue.

In this paper we formulate asking clarification question about user request in task-oriented dialogue based on task knowledge. To this end, we propose a dialogue-based user simulator and collect a novel dataset, called TaskClariQ, building on top of the ShARC (Saeidi et al., 2018) dataset. Our dataset includes a larger number of dialogue instances and has more complex task-related personalized information in user profiles. We propose a *System Ask* paradigm for response generation on task-oriented dialogues and propose a Multi-Attention Seq2Seq Network (MAS2S) architecture as an implementation of this paradigm, which generates clarification question and solutions in a single model. MAS2S comprises of a dialogue encoder, a user profile encoder, a task knowledge encoder, a solution confidence embedding network, and a response decoder. Experiments on TaskClariQ dataset demonstrate the effectiveness of MAS2S.

The contributions of this paper can be summarized as follows:

- We introduce the problem of asking clarifying questions on task-oriented dialogue based on user request, user profile, and task knowledge to better understand dialogue context.
- We design a dialogue-based user simulator to construct a new data collection called TaskClariQ for clarification question generation on task-oriented dialogues.
- We propose a System Ask paradigm for task-oriented dialogue and then propose a Multi-Attention Seq2Seq Networks (MAS2S) architecture as an implementation of this paradigm.

2 Related Work

2.1 Task-Oriented Dialogue

Task-oriented dialogue systems have focused on providing information and performing actions that can be handled by given task knowledge. Traditional systems (Wen et al., 2017; Eric et al., 2017; Lei et al., 2018; Zhong and Zettlemoyer, 2019; Liang et al., 2020; Feng et al., 2021; Yang et al., 2021) adopt a pipelined approach that requires dialogue state tracking for understanding the user's goal, dialogue policy learning for deciding which system action to take, and natural language generation for generating dialogue responses.

With the emergence of multi-domain Task-oriented dialogue datasets (Budzianowski et al., 2018; Shah et al., 2018; Rastogi et al., 2020; Feng et al., 2020; Gunasekara et al., 2021), the methodology is roughly seen to gradually progress from modularized modeling to generation and end-to-end modeling over the recent years. (Budzianowski and Vulić, 2019) first applied the GPT-2 model for the response generation task. (Lin et al., 2020) and (Yang et al., 2021) moved one step forward and utilized an end-to-end framework to solve task-oriented dialogue sub-tasks conditioned on the history of dialogue states. Based on the GPT-2 model, (Hosseini-Asl et al., 2020) proposed a cascaded model without using the oracle information. To improve the system performance, (Peng et al., 2021) and (Liu et al., 2021) applied dialogue pre-training over external dialogue corpora.

However, one of the major factors affecting task-oriented dialogue research is the lack of large-scale task-oriented dialogue data on the general domains. In addition, we noticed that task-oriented dialogue can be very personalized. Different users may need different solutions even on the same request. System should proactively ask questions of the users to clarify their personalized information needs. As a result, we collect a task-oriented dialogue dataset that contains clarification questions in the general domains, and we further propose an attention-based seq2seq model for clarification question generation.

2.2 Clarification Question Generation

With the emerging of various conversational devices, clarification question generation has achieved new attention in recent years. (Xu et al., 2019) collected a clarification dataset to address ambiguity arising in knowledge-based question answering. (Aliannejadi et al., 2019) proposed a clarification model to improve open-domain information-seeking conversations. (Kumar and Black, 2020) generated clarification questions by sampling comments from StackExchange posts. (Zhang et al., 2018; Rao and Daumé III, 2019) proposed an RL-based model for generating a clarifying question in order to identify missing information in product descriptions. (Cao et al., 2019) proposed to feed expected question specificity along with the context to generate specific as well as generic clarifying questions.

In contrast to prior work on clarification question, this work focuses on generating clarification questions to understand user request, user profile, and complex task-related dialogue context based on task knowledge in task-oriented dialogue system.

3 Problem Formalization

3.1 The System Ask Paradigm

An important challenge of a task-oriented dialogue system is that the system asks clarification questions to the users in order to understand the user's requests more accurately, and to increase its confidence with the provided solution. Based on this philosophy, we propose a clarification question generation paradigm in task-oriented dialogue as shown in Figure 2.

After a user initiates a dialogue session by providing an initial user request related to a task, the system generates a response with the *clarification question turn detection* module based on the user request, the user profile, and the task knowledge. If the system is not sufficiently confident with the generated solution, it will then generate a clarification question to ask using the *clarification question generation* module, which also considers the user request, the user profile, and the task knowledge. After the user responses to the clarification question, the system returns to the previous state, but this time it does not only consider the user's initial request but also the newly collected question-answer pair. This process will continue until the system is confident enough about the provided solution, in this case the system will display the solution to the user.

3.2 Notations and Problem Statement

Figure 1 shows an example of a task-oriented dialogue. A user has an initial user request R that relates to a specific task. In addition, a natural language description of the user profile U is provided.

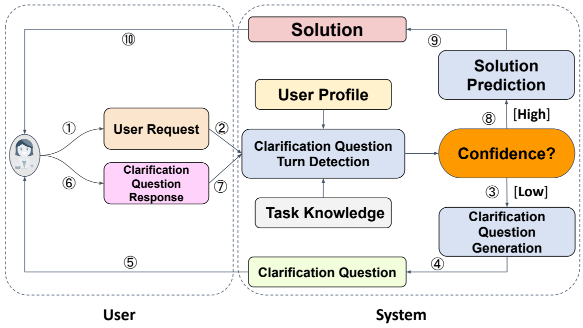

Figure 2: The workflow of the System Ask Paradigm.

We assume that R be solved using a snippet text K representing the task knowledge. If the user request R and user profile U are underspecified, i.e., the system cannot solve R directly and further information is required, the system needs to use the task knowledge T and user profile U to infer a clarification question Q in order to provide a more accurate solution to Y. We thus build the following conversation for this task-oriented behavior,

$$R|Q_0, A_0, Q_1, A_1, ..., Q_K, A_K|Y, \quad (1)$$

where $Q_k(0 \leq k \leq K)$ is the clarification question asked by the system, and $A_k(0 \leq k \leq K)$ is the response from user.

Based on the above notation, the task-oriented dialogue system aims at learning models for the following two key tasks:

Clarification Question Generation. Given a user request, a user profile, a task knowledge, and a dialogue history, generate the next clarification question to ask. Specifically, a generative model is trained by maximizing the probability of each clarification question in each of the training conversations:

$$P(Q_{k+1}|R, Q_0, A_0, \ldots, Q_k, A_k, U, T), \quad k \in \{0, \ldots, K\} \quad (2)$$

Solution Prediction. Given a user request, a user profile, a task knowledge, and a dialogue history, generate a solution for the user request. Specifically, a generation model is trained by maximizing the probability of the ground truth solution for each of the training conversations:

$$P(Y|R, Q_0, A_0, ..., Q_K, A_K, U, T) \quad (3)$$

Set	#Dialogue	#Task Knowledge	#User Profile	#Turns	#Tokens	Avg. turns per dialogue	Avg. tokens per turn
All	108,599	1,742	85,749	260,924	1,053,504	2.40	4.37
Training	76,019	687	55,048	184,027	733,413	2.42	3.98
Validation	10,860	495	10,545	25,473	105,071	2.34	4.12
Testing	21,720	560	20,156	51,424	215,020	2.36	4.18

Table 1: Number of dialogues, turns, task knowledge, user profiles, average number of turn per dialogue and average number of token per turn in training set, validation set and testing set of TaskClariQ.

4 Data Collection and Expansion

In this section, we explain how we built TaskClariQ dataset, that is, to the best of our knowledge, the first large dataset for task-oriented dialogue dataset with a focus on asking clarification questions.

We have built TaskClariQ on top of the ShARC[2] (Saeidi et al., 2018) dataset. The ShARC dataset is provided for conversational machine reading. This includes $32k$ question answering instances. However, some of the instances miss the answer to users' questions. It also lacks of task-related personalized information in user profiles. To this end, we build TaskClariQ, which includes $110k$ dialogues. Moreover, we added tasks-related personalized information in user profiles, which makes user profiles more personalized and related to the task. As such, we constructed TaskClariQ following a three-step strategy as follows:

4.1 Task-related User Profile Generation

Due to user profile in original ShARC dataset contains limited task-related personalized information, we first generate task-related dialogue to make user profile more related to task knowledge. We extract all unique clarification questions from all existing questions in the dialogues from the ShARC dataset. Then, we generate the task-related user profile based on their short answers (Yes/No) and the clarification question itself. To this end, we proposed a template-based approach to identify the type of clarification question, the verb, and the subject of the clarification question then we generate the task-related user profile. For instance, a question like "Are you a family farmer or fisherman?" with the answer "No", the type of question is "ARE", the verb is "Are", and the subject is "You", the task-related user profile is: "I am not a family farmer or fisherman.". Another challenge here is that some of the clarification questions can be answered in more than one way. Some questions use "AND" or "OR" statements, e.g., "Are you a fisherman or a sailor?". An OR (AND) question can have several positive and negative answers. Given the high complexity of these questions, we appointed three expert annotators for this task. Annotators needed to write all possible positive answers and negative answers for "OR" questions and "AND" questions.

4.2 Generated User Profile Verification

In this step, we aim to address the main concern which is how good are the generated user profile. To improve the quality of generated user profile, we also instructed the three annotators to read all the clarification questions and generated user profiles, correcting invalid and duplicate user profile.

4.3 From User Simulator to Dialogue Generation

Finally, in the third step, we propose a user simulation strategy to generate new dialogues and add generated task-related personalized information in the user profiles. For each dialogue, we have new generated task-related user profiles, which can be used to simulate a user and generate new dialogues. Specifically, we first add all possible new task-related user profiles by permuting the original user profile, and then remove the related clarification questions in the dialogue context. The outcome of this step is a new large set of conversations which makes the dataset larger, including a large pool of clarification questions. In addition, user profiles contain more task-related personalized information, which can better verify the clarification question generation ability for task-oriented dialogue systems.

We split the generated dataset into train, development, and test sets such that the train set includes 70% of the conversation, the development set contains 10% of them, and the rest 20% is the test set. Further, the details of TaskClariQ dataset composition can be seen in Table 1.

[2] https://sharc-data.github.io/data.html

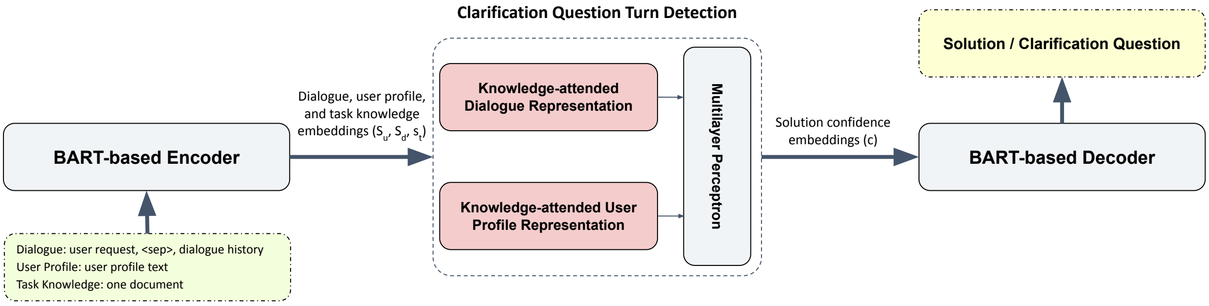

Figure 3: The Multi-Attention Seq2Seq Networks (MAS2S) architecture for task-oriented dialogue system.

5 Multi-Attention Seq2Seq Networks

In this section, we propose a task-oriented dialogue system that is able to ask clarification questions based on System Ask paradigm, which can provide solutions according to user request, user profile, task knowledge, and the dialogue history. Our approach MAS2S formalizes clarification question generation and solution prediction in task-oriented dialogue as a sequence to sequence problem using BART (Lewis et al., 2020) and Attention Networks (Vaswani et al., 2017). As shown in Figure 3, MAS2S consists of a dialogue encoder, a user profile encoder, a task knowledge encoder, a solution confidence embeddings network, and a response decoder. In each turn of dialogue, the dialogue encoder transforms the user request and all the dialogue history into the dialogue embeddings using BART encoder; the user profile encoder transforms the user descriptions into the user embeddings using BART encoder; the task knowledge encoder transforms the task rules into the knowledge embeddings also using BART encoder; the solution confidence embeddings network creates knowledge-aware dialogue representations and knowledge-aware user representations using attention mechanism to calculate solution confidence embeddings; finally, the response decoder sequentially generates a clarification question or a solution on the basis of the solution confidence embeddings using BART decoder.

5.1 Dialogue Encoder

The dialogue encoder takes the user request as well as all the dialogue history (user and system utterances) as input and employs BART to construct the dialogue embeddings. The relations between the user request and the dialogue history are captured by the encoder.

More specifically, to generate the semantic embeddings of dialogue, a BART encoder is given the token sequence $X = ([\texttt{CLS}], x_1, ..., x_N, [\texttt{SEP}], x_1, ..., x_M, [\texttt{CLS}])$, which are the sub-word tokens of user request and all the dialogue history respectively. [CLS] and [SEP] are start-of-text/end-of-text and separator pseudo-tokens. The output embeddings of each token is used as the dialogue semantic embeddings, referred to as $S_d = (d_1, ..., d_{N+M+3})$.

5.2 User Profile Encoder

The user profile encoder takes the descriptions of user scenario (a sequence of tokens) as input and employs BART to construct the user embeddings.

The input of the BART encoder is a sequence of user profile tokens with length N_u, denoted as $X = ([\texttt{CLS}], x_1, ..., x_{N_u}, [\texttt{CLS}])$, where [CLS] is start-of-text/end-of-text pseudo-tokens. The output is a sequence of embeddings with length $N_u + 2$, denoted as $S_u = (u_1, ..., u_{N_u+2})$ and referred to as user profile embeddings, with one embedding for each token.

5.3 Task knowledge Encoder

We also use a BART encoder to generate representations for task knowledge. It takes the rule text of task knowledge (a sequence of tokens) as input and output the task knowledge embeddings.

The input of the BART encoder is a sequence of task knowledge tokens with length N_t, denoted as $X = ([\texttt{CLS}], x_1, ..., x_{N_t}, [\texttt{CLS}])$, where [CLS] is start-of-text/end-of-text pseudo-tokens. The state of the final [CLS] is used as the task knowledge semantic embeddings, referred to as s_t.

5.4 Solution Confidence Embeddings Network

The solution confidence embeddings network takes the sequence of dialogue embeddings, the sequence of user profile embeddings, and the task knowledge embeddings as input and first calculates knowledge-attended dialogue representations and knowledge-attended user profile representations. In this way, the semantic information from dialogue context and user profile is represented based on task knowledge. Then solution confidence embeddings can be obtained by the reconstructed knowledge-attended semantic embeddings.

Specifically, we first use the attention mechanism to calculate knowledge-attended representations between task knowledge s_t and the dialogue S_d / user profile S_u by bilinear interaction, as follows:

$$A_d = \text{softmax}(\exp(S_d^{\mathsf{T}} W_d s_t)), \qquad (4)$$

$$A_u = \text{softmax}(\exp(S_u^{\mathsf{T}} W_u s_t)), \qquad (5)$$

where W_d and W_u are the bilinear interaction matrix to be learned. Then the knowledge-attended dialogue representations and knowledge-attended user profile representations are calculated as $d = S_d^{\mathsf{T}} A_d$ and $u = S_u^{\mathsf{T}} A_u$, respectively.

To obtain the solution confidence embedding c for current dialogue and user, we concatenate the knowledge-attended dialogue representations and knowledge-attended user profile representations. The solution confidence embedding c is derived by a multi-layer perceptron by the following equation:

$$c = \text{MLP}([d; u]). \qquad (6)$$

5.5 Response Decoder

The system response decoder generates the response by attending to the solution confidence embeddings. We employ a BART decoder for the system response decoder, which takes the solution confidence embedding c as its initial hidden state. At each decoding step t, the decoder receives the embedding of the previous item w_{t-1}, and the previous hidden state h_{t-1}, and produces the current hidden state h_t:

$$h_t = \text{BART}(w_{t-1}, h_{t-1}). \qquad (7)$$

A linear transformation layer is used to produce the generated token distribution p_t over the vocabulary:

$$p_t = \text{softmax}(V W_v h_t + b_v), \qquad (8)$$

where V is the token embedding of the collection of vocabulary for clarification question generation and the candidate solutions for user request, W_v and b_v are transformation parameters. During decoding, the decoder employs beam search to find the best sequences of tokens in terms of probability of sequence.

5.6 Training

The training of MAS2S follows the standard procedure of sequence-to-sequence. The BART model is fine-tuned in the training process. Cross-entropy loss is utilized to measure the loss of generating system responses.

6 Experiments

6.1 Datasets

We evaluate our models on TaskClariQ, our new collected dataset. It contains up to $110k$ dialogues consisting of a user profile, a task knowledge, a user request, a clarification question, and a user response. Each user profile is associated with task knowledge and includes more complex task-related personalized information. Table 1 provides some statics about this dataset.

6.2 Baselines

We compare between our approach and the state-of-the-art baselines in task-oriented dialogues.
Seq2Seq (Gu et al., 2016): a neural network-based Seq2Seq learning with copying mechanism, which can choose sub-sequence in the input sequence and put them at proper places in the output sequence.
SOLOIST (Peng et al., 2021): a transformer-based auto-regressive language model, which subsumes different dialogue modules into a single neural model to generate system responses for task-oriented dialogue system. We ignore the dialogue state and database information in our experiment.
UBAR (Yang et al., 2021): Fine-tuning the large pre-trained unidirectional language model GPT-2 to generate response on the sequence of the entire task-oriented dialogue session.

6.3 Evaluation Measures

We use the following evaluation metrics:
BLEU-X (Papineni et al., 2002): BLEU-X estimates a generated response's via measuring its n-gram precision against the ground truth. X denotes the maximum size of the considered n-grams (i.e. unigrams, bigrams, trigrams, and 4-grams).
ROUGE-X (Lin, 2004): ROUGE-X measures n-gram recall between generated and ground truth

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
Seq2Seq	0.217	0.131	0.046	0.036	0.221	0.075	0.211
SOLOIST	0.223	0.129	0.052	0.034	0.246	0.079	0.218
UBAR	0.274	0.165	0.086	0.048	0.291	0.102	0.273
MAS2S	**0.309**	**0.183**	**0.102**	**0.057**	**0.318**	**0.137**	**0.294**

Table 2: Performance of MAS2S and baselines on clarification question generation; Numbers in **bold** denote best results in that metric.

Model	Precision	Recall	F1	Accuracy
Seq2Seq	0.554	0.358	0.434	0.327
SOLOIST	0.572	0.353	0.436	0.352
UBAR	0.583	0.371	0.453	0.397
MAS2S	**0.604**	**0.418**	**0.494**	**0.412**

Table 3: Performance of MAS2S and baselines on solution prediction; Numbers in **bold** denote best results in that metric.

response. ROUGE-L measures the longest common word subsequence.

Solution Accuracy: The percentage of dialogues for which the solution is correctly identified.

Solution F1: F1 score of solution prediction, which includes precision and recall.

6.4 Implementation Details

We use a pre-trained BART-base model to encode dialogue, user profile and task knowledge. The max sentence length is set to 100. The hidden size of attentions are all set to 768. We also use beam search for decoding, with a beam size of 5. The dropout probability is 0.1. The batch size is set to 4. We optimize with Adam (Kingma and Ba, 2014) and an initial learning rate of 1e-4.

6.5 Experimental Results

Table 2 and Table 3 show the experimental results. We can see that MAS2S performs significantly better than the baselines in both clarification question generation and solution prediction. The results indicate that MAS2S is really a general model for task-oriented dialogue, which can effectively leverage the relation between dialogue, user profile, and task task knowledge to generate system response. We conjecture that the success of MAS2S is due to its suitable architecture design with BART-based encoder, confidence embeddings network, and BART-based decoder.

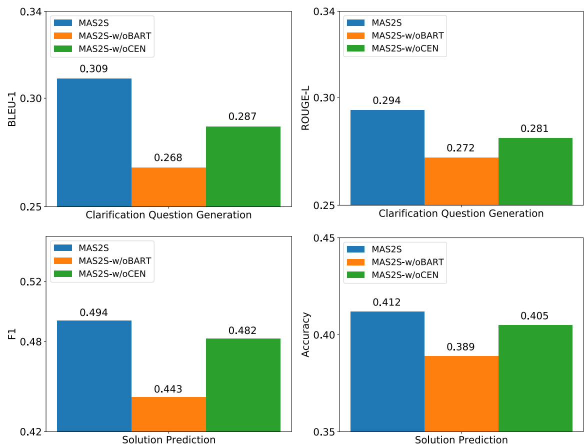

Figure 4: Ablation study results of MAS2S with respect to BART, and confidence embeddings network on TaskClariQ.

7 Discussions

7.1 Ablation Study

We also conduct ablation study on MAS2S. We validate the effects of two factors: BART-based encoder/decoder and confidence embeddings network. The results indicate that all the components of MAS2S are indispensable.

Effect of BART. To investigate the effectiveness of using BART in the dialogue encoder, user profile encoder, task knowledge encoder, and response decoder, we replace BART with Bi-directional LSTM and run the model on TaskClariQ. As shown in Figure 3, the performance of the BiLSTM-based model MAS2S-w/oBART in terms of BLEU-1, ROUGE-L, Accuracy, and F1 decreases significantly compared with MAS2S. It indicates that the BART-based encoder/decoder can create and utilize more accurate representations for dialogue, user profile, and task knowledge.

Effect of Confidence Embeddings Network. To investigate the effectiveness of using the confidence embeddings network, we compare MAS2S with MAS2S-w/oCEN which eliminates the confidence embeddings network module. Figure 3

Task Domain Knowledge	
Businesses that use Centrepay need to: 1. registration; 2. licensing and accreditation; 3. financial and privacy laws; 4. layby services.	
User Profile	
I follow all financial and privacy laws. My child is not isolated from school. We live in Philadelphia.	
Initial Request	
U: Can my Business use Centrepay?	
Dialogue Context	
S: Do you have licensing and accreditation?	
U: Yes	
S: Do you use layby services?	
U: Yes	
Generated response	
Ground Truth	Does your business have registration?
MAS2S	Do you register?
Seq2Seq	Do you follow financial and privacy laws?

(a) Example 1.

Task Domain Knowledge	
You'll get Cold Weather Payments if you get Universal Credit, and one of the following apply: 1. you get a limited capability for work amount; 2. you get the disabled child amount; 3. you have a child under 5 living with you.	
User Profile	
I am unemployed and my EHC is still valid.	
Initial Request	
Can I get Cold Weather Payments?	
Dialogue Context	
S: Do you have a child under 5 living with you?	
U: No	
S: Do you get the disabled child amount in your claim?	
U: No	
Generated response	
Ground Truth	Do you get the limited capability for work?
MAS2S	Do you have the capability for work?
Seq2Seq	No

(b) Example 2.

Figure 5: Case study on MAS2S and Seq2Seq on TaskClariQ. The generated response in green is a correctly predicted one, while the generated response in red is an incorrectly predicted one. The reason for generation is grounded to text in task knowledge and user profile in the same color.

shows the results on TaskClariQ in terms of BLEU-1, ROUGE-L, Accuracy, and F1. From the results we can see that without confidence embeddings network the performances deteriorate considerably. We conjecture that this is due to the attention mechanisms focused on task knowledge learn better semantic embeddings of dialogue and user profile. Therefore, MAS2S provides a more accurate indication of asking clarification question or providing solution to users.

7.2 Case Study

We make qualitative analysis on the results of MAS2S and Seq2Seq baseline on TaskClariQ. We find that MAS2S makes more accurate response by leveraging the relation existing in the dialogue, user profile and task knowledge. For example, in the first case in Figure 5, the user profile mentions that *"I follow all financial and privacy laws"*. MAS2S can correctly infer that system needs to ask clarification question about *"registration"* instead of *"financial and privacy laws"*. In the second case, the system needs to confirm whether the user *"gets a limited capability for work amount"*. MAS2S can effectively extract the relation between dialogue, task knowledge, and user profile, yielding a correct result. In contrast, Seq2Seq does not model the relations accurately and represent the confidence of the solution prediction. Thus it cannot properly generate the system response.

8 Conclusion

In this work, we introduced the task of asking clarification questions in task-oriented dialogue. We proposed a dialogue-based user simulator to construct and release a new data collection called TaskClariQ. We proposed a System Ask paradigm towards task-oriented dialogue. Based on this paradigm, we further proposed a Multi-Attention Seq2Seq Networks (MAS2S) as well as its solution confidence embedding network, which integrates the power of both sequential modeling and attention mechanisms. Experiments on TaskClariQ verified the performance of our approach against state-of-the-art task-oriented dialogue baselines. The research on asking clarification questions in task-oriented dialogue is still in its initial stage, and this work is just one of our first steps. In the future, the proposed paradigm may also be extended to more complex scenarios, such as considering task relation, dialogue relation, multimodal, etc.

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In SIGIR, pages 475–484.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. In CoRR.

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, pages 15–22.

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 EMNLP, pages 5016–5026.

Yang Trista Cao, Sudha Rao, and Hal Daumé III. 2019. Controlling the specificity of clarification question generation. In Proceedings of the 2019 Workshop on Widening NLP, pages 53–56.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, pages 37–49.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. Doc2dial: A goal-oriented document-grounded dialogue dataset. In EMNLP, pages 8118–8128.

Yue Feng, Yang Wang, and Hang Li. 2021. A sequence-to-sequence approach to dialogue state tracking. In ACL.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In ACL, pages 1631–1640.

Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, et al. 2021. Overview of the ninth dialog system technology challenge: Dstc9. AAAI.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In CoRR.

Vaibhav Kumar and Alan W Black. 2020. Clarq: A large-scale and diverse dataset for clarification question generation. In ACL, pages 7296–7301.

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 ACL, pages 1437–1447.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In ACL, pages 7871–7880.

Weixin Liang, Youzhi Tian, Chengcai Chen, and Zhou Yu. 2020. Moss: End-to-end dialog system framework with modular supervision. In AAAI, volume 34, pages 8327–8335.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, Barcelona, Spain. ACL.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In EMNLP, pages 3391–3405.

Qi Liu, Lei Yu, Laura Rimell, and Phil Blunsom. 2021. Pretraining the noisy channel model for task-oriented dialogue. Transactions of the Association for Computational Linguistics.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In COLING, pages 480–496.

Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Learning knowledge bases with parameters for task-oriented dialogue systems. In EMNLP, pages 2372–2394.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In ACL.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. Transactions of the Association for Computational Linguistics, 9:907–824.

Sudha Rao and Hal Daumé III. 2019. Answer-based adversarial training for generating clarification questions. In NAACL, pages 143–155.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In AAAI, volume 34, pages 8689–8696.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In EMNLP, pages 2087–2097.

Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In NAACL, pages 41–51.

Swadheen Shukla, Lars Liden, Shahin Shayandeh, Eslam Kamal, Jinchao Li, Matt Mazzola, Thomas Park, Baolin Peng, and Jianfeng Gao. 2020. Conversation learner-a machine teaching tool for building dialog managers for task-oriented dialog systems. In ACL, pages 343–349.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In NIPS, pages 5998–6008.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, 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 ACL, pages 438–449.

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. 2019. Asking clarification questions in knowledge-based question answering. In EMNLP-IJCNLP, pages 1618–1629.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In AAAI, volume 35, pages 14230–14238.

Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In CIKM, pages 177–186.

Victor Zhong and Luke Zettlemoyer. 2019. E3: Entailment-driven extracting and editing for conversational machine reading. In ACL, pages 2310–2320.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. In ACL, pages 142–149.