AirDialogue: An Environment for Goal-Oriented Dialogue Research

Wei Wei[1], Quoc V. Le[2], Andrew M. Dai[2] and Li-Jia Li[1]

[1]Google Cloud AI
[2]Google Brain
wewei, qvl, adai, lijiali@google.com

Abstract

Recent progress in dialogue generation has inspired a number of studies on dialogue systems that are capable of accomplishing tasks through natural language interactions. A promising direction among these studies is the use of reinforcement learning techniques, such as self-play, for training dialogue agents. However, current datasets are limited in size, and the environment for training agents and evaluating process is relatively unsophisticated. We present AirDialogue, a large dataset that contains 402,038 goal-oriented conversations. To collect this dataset, we create a context-generator which provides travel and flight restrictions. We then ask human annotators to play the role of a customer or an agent and interact with the goal of successfully booking a trip given the restrictions. Key to our environment is the ease of evaluating the success of the dialogue, which is achieved by using ground-truth states (e.g., the flight being booked) generated by the restrictions. Any dialogue agent that does not generate the correct states is considered to fail. Our experimental results indicate that state-of-the-art dialogue models on the test dataset can only achieve a scaled score of 0.22 and an exact match score of 0.1 while humans can reach a score of 0.94 and 0.93 respectively, which suggests significant opportunities for future improvement.

1 Introduction

Designing machines to talk like a human is one of the most important goals of research in machine learning and natural language generation. (Turing, 1950; Levin et al., 1997, 2000; Banchs and Li, 2012). Rooted in seq2seq models (Sutskever et al., 2014; Cho et al., 2014), recent neural based dialogue models (Shang et al., 2015; Sordoni et al., 2015; Vinyals and Le, 2015; Li et al., 2016a; Wen et al., 2016; Bordes et al., 2017; Lewis et al., 2017; Pieraccini et al., 2009; Serban et al., 2017) have generated promising results. However, building a robust and reliable agent that can hold a conversation with humans while achieving a specific goal remains an open challenge. While a majority of previous work studied *chitchat models* (Ghazvininejad et al., 2018; Sordoni et al., 2015), in this paper we focus on *goal-oriented models* (Li et al., 2017; Liu and Lane, 2017; Liu et al., 2017) for conversations.

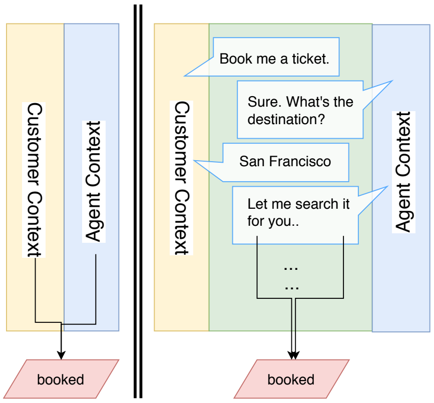


Figure 1: Goal-Oriented Dialogue Environment. (left) Context pairs are mapped to unique states in the environment. (right) Conversation models can only access its own private context and utterance in the public domain. At the end of the conversation, dialogue states are generated from one of the agents using information from the utterance.

We define a goal-driven dialogue to be a conversation that is conditioned on a pair of contexts $c = (u, v)$, with the goal of reaching the target state $s \in \mathcal{S}$. For a dialogue environment E, there exists a mapping f_E that maps from the context pair to the target state (i.e., $s = f_E(c)$). While an environment can access to the full context pair, a dialogue agent, say A_u, can only access its own private context u and the dialogue history $h_t = \{x_1, x_2, ...x_t\}$ with x_t being an utterance generated by one of the agents at time t (i.e., $x_{t+1} \sim A_v(x|v, h_t)$

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3844–3854
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

or $x_{t+1} \sim A_u(x|u, h_t)$). By forbidding accessing to the context of the other party, goal-driven dialogues will have to be developed so that the dialogue history h_t contains all the information that is necessary for a particular agent, say A_v to reach the target state s of the conversation defined by the environment through a mapping g_v (e.g., $s = g_v(h_t, c_v) = f_E(c)$). When one of u or v is a human, A_u and A_v will have to belong to a class of generators that respond in natural language.

We present AirDialogue, a large-scale corpus with 402,038 dialogues and an environment that makes it easy to simulate and evaluate goal-oriented dialogues. Our setting is centered around the theme of a flight booking session between a customer and a support agent. Since it's easy to find a rule based strategy to book a ticket given all the constraints, a mapping can be easily found in order to generate the ground-truth state (e.g., the ticket that needs to be booked) for each dialogue context so that we can evaluate the generated dialogue. In our environment, a context pair c always comes with a unique s. If the dialogue agent generates a state s' that is different from s, the agent has failed to achieve the goal. We use this as a mechanism to measure the performance of dialogue agents. We consider an additional metric to measure the "natural languageness" of the conversations so that the agents do not just exchange bits.

We have implemented some strong dialogue generation models and experimented with them on our dataset. Experimental results demonstrate that even the most advanced model can only achieve a benchmark score of 0.33. Comparing that to the human score of 0.94, that leaves for significant future improvement.

2 Existing Datasets

A comparison between the AirDialogue and several publicly available ones is shown in Table 1. Existing datasets are usually too small to support deep learning approaches to model dialogues generation. As a comparison, the WMT'15 English-Czech dataset (Luong and Manning, 2016), a benchmark dataset for machine translation, contains 15.8 million translation pairs whereas the current largest goal-oriented dataset has only 20,300 conversations. Synthesized data can also be an option to obtain a large dataset. However, these are often built from templated responses which make it meaningless for dialogue models to learn. Another issue with conversation datasets is the lack of a sophisticated environment that can be used to evaluate a generated dialogue. Some of the recent datasets provide an environment but are generally not representative enough to model real-world settings as illustrated by a narrow context space. As a result, the limited availability of datasets and complex environments have become a bottleneck for research in goal-oriented dialogue.

Our dataset has more than 20 times as many samples as found in the biggest of the existing public datasets. In addition to the number of samples, we have also compared the context complexity and the state complexity. Context complexity measures the unique number of context that a conversaion can be grounded into and state complexity measures the number of states that a conversation can reach. As we can see from the table, AirDialogue has the largest complexity in both context and state, giving it the flexibility to form a diverse selection of goal-oriented conversations. Our dataset also supports a wide range of tasks that can be found in the dialogue community. These include dialogue generation, state tracking and dialogue self-play.

3 Task Environment

We formulate the flight booking problem as a collaborative goal-driven dialogue problem that was defined in the introduction. Two types of agents are present: customers and agents. Dialogues are conditioned on a context pair $c = (c_c, c_a)$, with c_c being the context for the customer and c_a for the context of the agent. Here, the customer context $c_c = (tr, o)$ consists of the goal of the dialogue o (i.e., book, change or cancel) as well as the travel constants tr. Agent context $c_a = (db, r)$ consists of available flights in the database db and a field r indicating whether the customer has an existing reservation in the system. A final dialogue state s is derived at the end of the conversation once the agent has acquired all the information and the customer has confirmed all the changes in their reservation.

Task Logic. One of the main purposes of the flight booking problem is to mix decision making in the context of a dialogue. Figure 2 illustrates the task logic in order to successfully solve our problem. The goal of the conversation is provided as part of the customer's context, which has to be one of the following:

- book: make a new reservation

Dataset	Context complexity	State complexity	Supported tasks	Num. samples
Real Datasets				
AirDialogue	$\geq 4.43 \times 10^{178}$ [a]	750,000 [b]	Dialogue Generation Dialogue Self-play State Tracking	402,038
DSTC1-4 (Henderson et al., 2014)	Unknown	Unknown	State Tracking	20,300
Stanford CoCoA (He et al., 2017)	16	N/A	Dialogue Generation	11,157
Talk and Walk (de Vries et al., 2018)	80	2	Dialogue Generation	10,000
Negotiation Chatbot (Lewis et al., 2017)	3	7×3	Dialogue Generation Dialogue Self-play	5,808
Frames (El Asri et al., 2017)	Unknown	20	Dialogue Generation State Tracking	1,369
Key-Value Retrieval Networks (Eric et al., 2017)	284	284	Dialogue Generation State Tracking	3,031
Cambridge Restaurant System (Wen et al., 2016)	Unknown	Unknown	Dialogue Generation State Tracking	680
Synthesized Datasets				
AirDialogue Synthesized	$\geq 4.43 \times 10^{178}$	750,000	Dialogue Generation State Tracking Dialogue Self-play	-
Facebook bAbI dialog tasks (Bordes et al., 2017)	Unknown	Unknown	Dialogue Generation State Tracking	-
Task-Completion Dialogue Systems (Li et al., 2016b)	Unknown	3	Dialogue Generation State Tracking	-

[a] Calculated based on all possible combinations of customer and agent context features in Table 2 and Table 3. Assume 365 days a year, 24 airport codes, 8 airlines and 30 flights in the database with each flight having the same departure and arrival date as the intent and is always under the customer's budget. This is a conservative estimate since the actual dataset have flights with different dates and prices.

[b] Calculated based on 30 flights in the database, 5,000 names and 5 dialogue action states.

Table 1: Comparing AirDialogue to Existing Datasets

- change: change an existing reservation
- cancel: cancel an existing reservation.

The agent is then expected to follow the task logic and guide the conversation all the way to one of the five dialogue state actions. For example, when the goal o is "book", the agent will iterate through each of the customer's set of travel restrictions tr and search for available flights in db. If there are available flights, the conversation will be concluded with the status action "booked". Otherwise a status action of "no flight found" will be returned. On the other hand, the task logic for customers with a goal of "change" would be slightly different. Agents are supposed to check for r to determine whether a reservation exists. If it does, the agent will interact with the customer to update the travel constants tr. Otherwise, a status action will be selected with "no reservation". Similarly, the conversation will conclude "no flight found" if none of the flights in db satisfies the customers' need and "changed" if the the new flight is found. Finally, for customers who wish to cancel their ticket, the agents will perform a simple check and cancel if the reservation is found and "no reservation" otherwise.

Agent Context. There are two components in the agent context $c_a = (db, r)$. $db = (f_1, f_2, \ldots, f_m)$ is a list of flights each with 12 features listed in Table 3. Each feature has a prior distribution that we use to generate those settings. For example, 90% of the flights in the database would be economy class

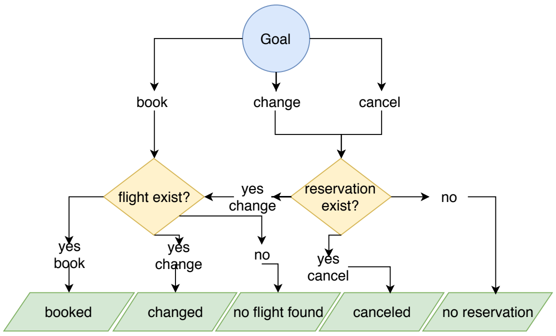


Figure 2: Task Logic of the flight booking problem

while 10% of the flights would be business class. The flight database is unique to each conversation. The price of the flights are drawn from a Gaussian distribution with mean μ and standard deviation $\sigma = \mu * \beta$. μ is 210 for economy class and 650 for business class. β is 0.2 for direct flights, 0.4 for flights with one connection and 0.6 for those with 2 connections. To simplify our setting, we only consider round trip flight tickets with both trips under the same airlines. r is simply a binary variable indicating whether the customer has previously made a reservation.

Customer Context. Customer context $c_c = (tr, o)$ also consists of two pieces. $tr = (tr_1, tr_2, \dots, tr_n)$ is a list of travel restrictions indicated in Table 2. Here we constrain the form of travel restrictions into the ones that are most useful for the flight booking situation, which is illustrated in Table 3. For example, customers may request a flight with either economy class, business class or accepts anything that is available. Some of the restrictions requires certain level of common sense knowledge to "translate" into an actual search query. Take travel time for example, a morning flight would corresponds the flight between 3am to 11am and a standard fare airline would be one of the big brand airline companies. The rest of the airlines are considered low-cost airlines. The probability of each occurrence that will be appeared in the customer context is also listed in the table.

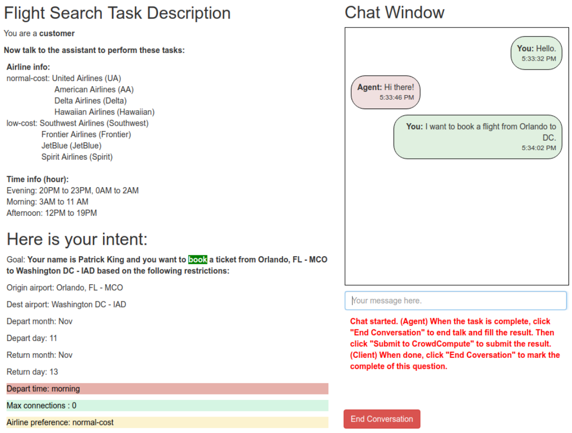
Dialogue States. At the end of the conversation, agent will submit the dialogue states $s = (s_a, s_n, s_f)$, a state action s_a which will be one of the following 5 : "booked", "changed", "no flight found", "no reservation" and "cancel", the name of the customer s_n and the flight being selected for this dialogue s_f. Flights will be identified by a flight number that indicates one of the m flights in the database.

Environment. As we discussed earlier in the introduction, there exists a mapping $f : c \rightarrow s$ so that we can acquire the final dialogue state directly from the context pair. This mapping corresponds to our environment and the expected state $s' = f(c)$ generated from the context pair can be used to evaluate the state s generated from our algorithm.

Sentence Level Annotation. In addition to dialogue context and states, some of the sentences in the dialogues are also labeled during the data collection process. The sentence level annotation records the items agent clicked on the web UI when we were collecting the dialogue data. Agents are given the instructions to input all the travel constraints immediately after they receive them from the customers via the chat window.

4 Datasets

In this paper we present the AirDialogue dataset that contains a large collection of human generated dialogues. In addition, we also present the syntherized dataset generated using a templated simulator, along with an out-of-domain dataset that contains context that drawn from a different prior distribution than the previous two. AirDialogue and the synthesized datasets are divided into train, dev and eval sets randomly by applying a ratio of 80%, 10% and 10%. Details of the statistics are shown in Table 5.


Flight Search Task Description

You are a **customer**

Now talk to the assistant to perform these tasks:

Airline info:
normal-cost: United Airlines (UA)
American Airlines (AA)
Delta Airlines (Delta)
Hawaiian Airlines (Hawaiian)
low-cost: Southwest Airlines (Southwest)
Frontier Airlines (Frontier)
JetBlue (JetBlue)
Spirit Airlines (Spirit)

Time info (hour):
Evening: 20PM to 23PM, 0AM to 2AM
Morning: 3AM to 11 AM
Afternoon: 12PM to 19PM

Here is your intent:

Goal: **Your name is Patrick King and you want to book a ticket from Orlando, FL - MCO to Washington DC - IAD based on the following restrictions:**

Origin airport: Orlando, FL - MCO

Dest airport: Washington DC - IAD

Depart month: Nov

Depart day: 11

Return month: Nov

Return day: 13

Depart time: morning

Max connections : 0

Airline preference: normal-cost

Chat Window

You: Hello.
5:33:32 PM

Agent: Hi there!
5:33:46 PM

You: I want to book a flight from Orlando to DC.
5:34:02 PM

Chat started. (Agent) When the task is complete, click "End Conversation" to end talk and fill the result. Then click "Submit to CrowdCompute" to submit the result. (Client) When done, click "End Coversation" to mark the complete of this question.

Figure 3: Customer's Interface

AirDialogue Dataset. To collect human annotated dialogue data, we first generate context pairs based on the prior distributions defined in Table 3

feature	goal	class	max. price	airline	dep./ret. time
condition (prob.)	book (80%) change (10%) cancel (10%)	economy (7%) business (3%) any (90%)	$\leq$200 (25%) $\leq$500 (25%) $\leq$1000 (25%) any (25%)	standard fare (5%) UA, Delta AA, Hawaiian any (95%)	morning (3%) 3am-11am afternoon (4%) 12pm-19pm evening (3%) 20pm-2am any (90%) 0am-23pm
feature	dep./ret. month	dep./ret. day	max. conn.	dep./ret. airport	
condition (prob.)	uniform	uniform	0 (7%) 1 (90%) any (3%)	uniform	

Table 2: List of Customer Travel Restrictions

feature	dep./ret.city	dep./ret. month	dep./ret. day	dep./ret. time
range	categorical	1-12	1-31	00-23
prob.	uniform	uniform	uniform	uniform
feature	class	price	connections	airline
range	business,economy	0-5000	0,1,2	categorical
prob.	10%, 90%	See Section 3	7%, 90%, 3%	uniform

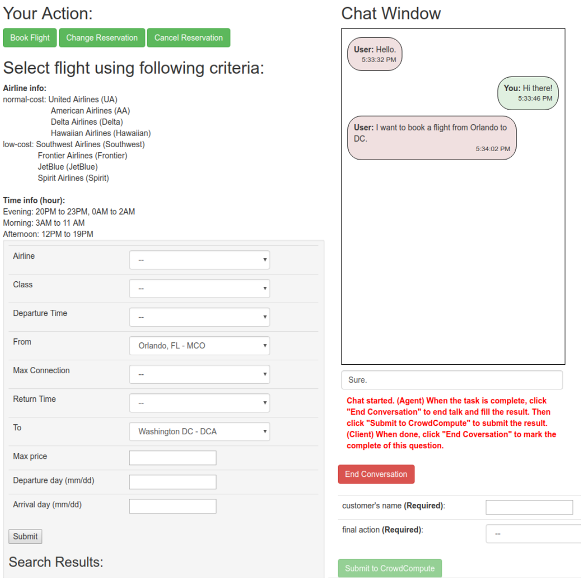
Table 3: Flight Features

avg. duration	4.7mins	vocab size	25,621
avg. dialogue len	115	avg. turns	14.10
avg. turn len	8.17	num. diag.	402,038
sent. annotation %	36.1	correct %	88.5

Table 4: Statistics of the AirDialogue Dataset

	Train	Dev	Test	Total
AirDialogue	321,460	40,363	40,215	402,038
Syntherized	320,000	40,000	40,000	400,000
OOD Context	-	40,000	40,000	80,000

Table 5: Statistics of All the Datasets Used in the Paper



Select flight using following criteria:

Airline info:
normal-cost: United Airlines (UA)
American Airlines (AA)
Delta Airlines (Delta)
Hawaiian Airlines (Hawaiian)
low-cost: Southwest Airlines (Southwest)
Frontier Airlines (Frontier)
JetBlue (JetBlue)
Spirit Airlines (Spirit)

Time info (hour):
Evening: 20PM to 23PM, 0AM to 2AM
Morning: 3AM to 11 AM
Afternoon: 12PM to 19PM

Airline	--
Class	--
Departure Time	--
From	Orlando, FL - MCO
Max Connection	--
Return Time	--
To	Washington DC - DCA
Max price	
Departure day (mm/dd)	
Arrival day (mm/dd)	

User: Hello.
5:33:32 PM

You: Hi there!
5:33:46 PM

User: I want to book a flight from Orlando to DC.
5:34:02 PM

Sure.

Chat started. (Agent) When the task is complete, click "End Conversation" to end talk and fill the result. Then click "Submit to CrowdCompute" to submit the result. (Client) When done, click "End Coversation" to mark the complete of this question.

customer's name **(Required)**:

final action **(Required)**: --

Figure 4: Agent's Interface

and Table 2. 30 flights are generated for each dialogue. Annotators are then asked to role play the dialogue using the web interface illustrated in Figure 3 and Figure 4. The customer is shown with the goal and any requirements, as well as the chat history. The agent has a similar interface with the addition of a search feature that will search and return the cheapest flights that satisfy the given search constraints. The layouts and colors of the UI were optimized to reduce human errors. Human annotators are highly familiar with the settings of the task as most of them stayed in the project full time for more than 6 month. A human project manager manually examines roughly 5%-6% of the data each day and provide feedbacks to the human annotators to ensure the quality of the data collection. Table 4 shows some of the statistics of the AirDialogue dataset. On average, 88.5% of the dialogues generated by human reaches a perfect state. In the next Section we will analyze the types of human mistakes. In addition to dialogue history, we have also recorded agent search events (e.g. adding a new search constant through the web UI) on each turn, which are sentence level dialogue state annotations. Annotators are given the instructions to put search constraints immediately after they have received them from the natural conversation. 36.1% of the dataset dialogues have access to such information. Tracking search events provides a structured representation of progress of the dialogue.

Synthesized Dataset. In addition to the AirDialogue dataset collected using human annotators, we have also built a dialogue simulator to generate synthesized dialogues. The dialogue simulator relies on the context generator with the same set of priors. Synthesized dialogues are generated by following a set of templates and alternate between them randomly.

Out-of-domain Context Set. We have also generated an out-of-domain context set that does not contain any dialogues. This context set is generated by setting the goal probability from the one showing in Table 2 to a uniform distribution. The reservation probability is also changed from 10% to 70%. The sets of customer name and airport codes have also gone up significantly in those two datasets. This makes it difficult for models with fixed vocabulary size to perform well on those OOD datasets.

5 Data Analysis

5.1 Required Skills

This dataset presents many challenges for existing methods. Table 6 lists some of the skills that are required to accomplish the flight booking task.

Lexical and Syntactic Variations. Human language is diverse and there are many forms of lexical and syntactic variations. Taking the examples in Table 6, the amount of variation that appears in human dialogue poses great challenges for conversational models.

Applying External Knowledge. Another challenge in our data set is the use of external (commonsense) knowledge. Vaguely defined concepts such as morning and afternoon are used comfortably by humans. However, a learning algorithm needs to successfully adapt those concepts when searching for flights. An alternative way to solve this problem is to inject external knowledge into the algorithm, which is ananother important issue in dialogue systems.

Active Information Seeking Conversation. We have observed that human annotators who have high correct rates often have the habit of actively requesting information. They take extra steps to ensure all the flight search conditions are correctly communicated. This is especially important since customers are the only party in the dialogue who have access to the travel restrictions.

Goal-driven Dialogue Development. Another necessary skill to solve the flight booking problem is to develop dialogue that can be used to drive the conversation towards its end goal. Having such a goal in mind distinguishes goal-oriented models from chitchat models and makes the conversation more effective and efficient.

Reasoning over Large Structured Data. Selecting flights relies on effective methods to reason over a large scale structured database. This is a challenge that has practical impact but has rarely been addressed in previous research.

Learning from Multiple Solutions. A final challenge in the problem is the fact that there exists multiple equally optimal flights to the same set of customer restrictions.

5.2 Analysis on Human Mistakes

As we have reported in Table 7, the human error rate on this task is close to 10%. We have analyzed the human errors and grouped them into 6 categories. Here an invalid status indicates that agents have chosen a status that they are not supposed to reach according to Figure 2. For example, a “book” goal should not reach “no reservation” as an action status. “Wrong status”, on the other hand, is a possible action status to reach but are not expected given the context of the conversation. Minor mistakes comprise of situations that include when agents misspell the name of the customer but get everything else correct. Those mistakes can be fixed in the dialogue from the ground truth. The majority (85%) of the errors happened when communicating flight search constraints, and entering wrong conditions that lead the search tool to return no results (6.8%).

6 Methods

6.1 Supervised Learning

Model Architecture. Our supervised dialogue model is built based on the seq2seq model(Sutskever et al., 2014). We treat both context from customer and agent as sequences and encode them using RNN. For customer context c_c we encode it using a single RNN. To encode agent context c_a we apply a hierarchical RNN structure by first encoding each flight using an RNN and then encode the outputs of each encoded flights along with the reservation information using another RNN. Utterance of time t is generated using

Skills	Examples
Lexical and Syntactic Variations	Customer: My travelling dates are **Aug 12-14**. Customer: I want to take off on **Sept 18** and please confirm my return ticket on **Sept 20**. Customer: Travelling dates are **12/13** and **12/15**.
Applying External Knowledge	Customer: I am traveling on Oct, 10 and I am returning on Oct, 12 in the **evening**. Customer: I prefer **normal-cost** airlines.
Active Information Seeking	Customer: And **please make sure** that the departure time is at afternoon. Agent: Do you have any **other specifications**? Agent: Can you **mention** Washington airport code for me? Agent: Do you like to travel in a **economy class or business class**?
Goal-driven Dialogue Development	Agent: Sure, please **provide me with your planned travel dates**. Agent: Hello, **how can I help you**? Agent: Thank you for reaching out to us. **Have a great time**.

Table 6: Conversational Skills required to accomplish the AirDialogue task.

invalid status	4.4%	minor mistakes	2.0%
reservation	1.0%	flight constant	85%
wrong condition	6.8%	wrong status	0.8%

Table 7: Human error statistics.

a sequence2sequence model by concatenating the context embedding along with the embeddings of conversation history h_{t-1}. Agent and customer will have their own model $P(x_t|h_{t-1}, c_a; \theta_a)$ and $P(x_t|h_{t-1}, c_c; \theta_c)$. At the end of the conversation the dialogue state will be generated in a sequence using another sequence2sequence model by taking the entire conversation history and the agent context, $P(s_i|s_{i-1}, h_T, c_a; \theta_s)$.

Optimization. During supervised learning, we optimize the model by considering the loss from both the dialogues x and their states s. A token x_t can belong to a either customer utterance ($x_t \in \pi_c$) or an agent utterance ($t \in \pi_a$). The parameters for supervised learning contains all the parameters of the models: $\Theta = \{\theta_a, \theta_c, \theta_s\}$. In supervised learning we optimize the following loss function.

$$\ell(\Theta) = -\sum_{(x,c,s)\in\mathcal{D}} \sum_{t=1}^{T} \mathbb{1}_{\pi_c}(x_t) \log P(x_t|h_{t-1}, c_c; \theta_c) + \\ \mathbb{1}_{\pi_a}(x_t) \log P(x_t|h_{t-1}, c_a; \theta_a) + \\ \sum_i \log P(s_i|s_{1:i-1}, h_T, c_a; \theta_s)$$

6.2 Reinforcement Learning Self-Play

Supervised learning for dialogue generation is known for many issues such as generating templated responses regardless of the inputs (Li et al., 2015). Here we design a reinforcement learning self-play algorithm to enable the model to learn from the environment by chatting with each other. Our self-play model is initialized using a model trained from the supervised learning. Since no conversation data is involved in the self-play, we generate context pairs directly from the context generator during training. Here we consider terminal rewards, which is generated by simulating the dialogue all the way to the end and compare the generated state s with the ground truth state s'. We use the scaled score as rewards introduced in the paragraph of Evaluation Metrics in Section 7.

Value Network. To reduce variance, we build a value network to provide a baseline estimate for returns. Both agent and customer gets their own value network $v_a(h_t|c_a; \theta_{v,a})$ and $v_c(h_t|c_c; \theta_{v,c})$. The value functions are parameterized by a seq2seq model and a linear transform applied on its output. During the training of the value functions, the main model parameters Θ are fixed and the only trainable variables are $\theta_v = \{\theta_{v,a}, \theta_{v,c}\}$.

Policy Network. We use the same structure as in supervised learning to be our policy network. We adopt REINFORCE algorithm (Williams, 1992) to optimize our algorithm using the following gradient.

$$\nabla \ell_{RL}(\Theta) = \\ \mathop{\mathbb{E}}_{x_t \in \pi_c} (R_t - v_c(h_{t-1})) \nabla \log P(x_t|h_{t-1}, c_c; \theta_c) + \\ \mathop{\mathbb{E}}_{x_t \in \pi_a} (R_t - v_a(h_{t-1})) \nabla \log P(x_t|h_{t-1}, c_a; \theta_a) + \\ \mathop{\mathbb{E}}_{s_i} (R_t - v_a(h_T, s_{i-1})) \nabla \log P(s_i|s_{i-1}, h_T, c_a; \theta_s)$$

Experiments	pplx	BLEU	Name Acc.	Flight Acc.	Action Acc.
Synthesized dev	1.08	68.72	100%	100%	100%
Synthesized test	1.08	68.73	100%	100%	100%
AirDialogue dev	2.21	23.26	100%	100%	100%
AirDialogue test	2.20	23.75	100%	100%	100%

Table 8: Dialogue Generation and State Prediction

Experiments	Name	Flight	State	Total	BLEU
Supervised (Synthesized dev)	0.39(0%)	0.11(8%)	0.32(32%)	0.23(0.14)	**68.72**
Self-play (Synthesized dev)	**0.47(0%)**	**0.36(35%)**	**0.39(39%)**	**0.39(0.29)**	62.71
Supervised (Synthesized test)	0.39(0%)	0.08(4%)	0.33(33%)	0.22(0.12)	**68.73**
Self-play (Synthesized test)	**0.47(1%)**	**0.35(16%)**	**0.47(47%)**	**0.41(0.22)**	62.66
Supervised (AirDialogue dev)	0.4(0.9%)	0.07(1.2%)	0.12(12%)	0.15(0.04)	**23.26**
Self-play (AirDialogue dev)	**0.41(1%)**	**0.13(4%)**	**0.29(29%)**	**0.23(0.11)**	19.65
Supervised (AirDialogue test)	0.39(1%)	0.08(1.6%)	0.08(8%)	0.14(0.03)	**23.15**
Self-play (AirDialogue test)	**0.43(1%)**	**0.11(3%)**	**0.28(28%)**	**0.22(0.10)**	18.84
Human (AirDialogue test)	1 (98%)	0.92 (91.4%)	0.92 (91.8%)	0.94 (0.93)	-

Table 9: Dialogue Self-play Displayed with Scaled Scores and Their Exact Match Scores in the Parentheses.

Exp	Name	Flight	State	Dialogue
OOD1	0.4(0%)	0.1(1%)	0.18(18%)	0.18(0.06)
OOD2	0.4(0%)	0.09(2%)	0.21(21%)	0.19(0.07)

Table 10: Performance of Trained Self-play Models on Out-of-domain Context Pairs using AirDialogue Data

7 Experiments

Experiment Setup. We implemented our model using Tensorflow using SGD as the optimizer with a learning rate of 0.1 and a batch size of 64. The seq2seq model was implemented using 4 layers of GRU with a hidden unit 384. Greedy Decoder is used for seq2seq decoding. Inputs are tokenized using NLTK [1]. For AirDialogue dataset, tokens occurred less than 10 times are eliminated but no tokens are removed for the synthesized dataset. As a result, there are 5,547 tokens left the experiments. There are 700 tokens for the synthesized dataset and no tokens are eliminated during the pre-processing. In training we only applied the dialogues that have correct states.

Accelerate Training In the usual seq2seq diagram for dialogue generation, one would treat a single conversation with k turns as k different training samples by feeding conversation before the k^{th} turn into the encoder and use a decoder to generate the k^{th} turn. Such a training strategy would encode the dialogue history repeatedly. We apply a technique to speed up training that is illustrated in Figure 5. Here the encoder is never needed to encode a single dialogue multiple times since its outputs are reused for multiple turn predictions. The decoder generates the output sequence by alternating its states between previous decoder state and the encoder states. If the sentence is within the boundary of the current turn, its hidden state got passed from its previous state. Otherwise, its hidden state will be "reset" into the corresponding state in the encoder. One can easily implement this training strategy and use a pre-processed Boolean array to represent whether a token is within a turn for a specific agent.

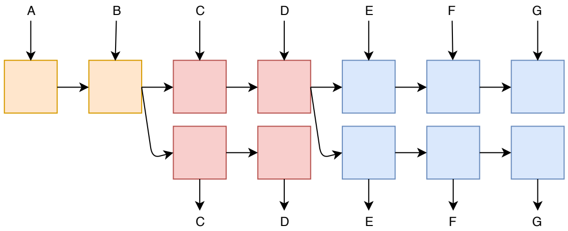


Figure 5: Techniques to speed up training. Here a conversation with 3 turns are annotated using colors. The encoder only needs to pass through the dialogue once for the entire dialouge sample to be trained.

Evaluation Metrics. We use perplexity and BLEU score to evaluate the quality of the language generated by the model. We also compare the dialogue state generated by the model s and the ground truth state s'. Two categories of the metrics are used: exact match scores and scaled scores. In an

[1] https://www.nltk.org

exact match metric, dialogue state is given a score of 1 if it matches exactly to the ground truth and 0 otherwise. In a scaled metric, scores are scaled between 0 and 1 to provide information that are of finer granularity. There are three dialogue states: name, flight and action. For name, scaled metric is chosen to be the character-wise F1 score. For flight, scaled metric is chosen to be 1 minus the scaled distance between the selected flight f and the ground truth F_g. Note there might be multiple optimal ground truth flights that have the same price and satisfy the customers' requirements. Therefore F_g should be a set of flights. The distance function $d(f1, f2)$ is a measure of distance on each of the flight features. The scaled score on flight is calculated as the following. Here F is a set of all flights in the datasbase.

$$score(f) = 1 - \frac{\min_{f' \in F_g} d(f, f')}{\max_{f1 \in F, f2 \in F_g} d(f1, f2)}$$

Dialogue action states can only have exact match metrics. Finally, the total score of a dialogue is taken to be a weighed sum scores of name, flight and dialogue status by a factor of 0.2, 0.5 and 0.3 for both scaled and discrete metrics.

Dialogue Generation and State Prediction. We train the models on the train sets and show their performs on the dev and test sets in Table 8. The BLEU score measured by comparing the generated response and the ground truth is around 68.7 for synthesized data and around 23 for AirDialogue. Given the fact that templated dialogues are easier to learn, it is expected that the synthesized dataset gets a high BLEU score. In the state prediction task, the model paper achieved a perfect accuracy across all the dialogue states given the ground truth dialogue and previous states. However, as we will see shortly, the triumph on ground truth hisotry might not be able to be transferred to self-play experiments, which generates dialogues that have different distributions from the ground truth data.

Dialogue Self-play. During the self-play experiments we perform similar predictions on the dialogue states. However, instead of asking the models to predict those states given ground truth history, we now ask the models to predict given the generated dialogues. Table 9 shows the results using both the supervised model and the self-play model. Here we see significantly improvements across all measures for self-play models compare to their supervised learning models. However, the fact that the exact match scores are so low indicates that our models are far from mastering the goal-oriented dialogue problem in the self-play setting as the rewards and accuracies are consistently low. As a comparison, human agents achieved nearly 90% on rewards across all categories, which sets a good target for future work in the field. One possible reason for the low exact match score but relatively high scaled score is because we use the scaled score as rewards in out self-play training. As a result, the metrics are highly tuned toward scaled scores instead of exact match scores. One can apply techniques such as pointer networks (Vinyals et al., 2015) which is possible to optimize exact match scores in a better way. To prevent language from degenerating into binary bits, we mix three supervised training steps on the train data with one reinforcement learning update during self-play training. By doing this, we are able to maintain a BLEU score at similar level compares to the supervised learning.

Out-Of-Domain Self-play. We have also conducted experiments on the out-of-domain context pairs. The results are shown in Table 10. The out-of-domain context pairs contain dialogue contexts with distribution far deviated from the training data. It is not surprised to see here that our model does not perform as good as in the testing data using the data it is familiar with.

8 Conclusions

In this paper, we propose an environment for goal-oriented dialogue research based on the problem of flight bookings. We have collected a dataset that is more than 400,000 conversations. Our environment allows easy generation of new dialogue contexts and allows verification of the generated dialogues, which can be used to support a wide range of research such as dialogue self-play. Although supervised learning seems to perform well in our setting, self-play poses a challenge for goal-oriented dialogue research. The gap between our self-play approach and the human baseline suggests possibilities for significant future improvements.

References

Rafael E Banchs and Haizhou Li. 2012. Iris: a chat-oriented dialogue system based on the vector space model. In *Proceedings of the ACL 2012 System Demonstrations*, pages 37–42. Association for Computational Linguistics.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *AAAI Conference on Artificial Intelligence*.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1766–1776, Vancouver, Canada. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 1997. Learning dialogue strategies within the markov decision process framework. In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 72–79. IEEE.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1):11–23.

Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016a. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202. Association for Computational Linguistics.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016b. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. *arXiv preprint arXiv:1708.05956*.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2017. End-to-end optimization of task-oriented dialogue model with deep reinforcement learning. *arXiv preprint arXiv:1711.10712*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063. Association for Computational Linguistics.

Roberto Pieraccini, David Suendermann, Krishna Dayanidhi, and Jackson Liscombe. 2009. Are we there yet? research in commercial spoken dialog systems. In *International Conference on Text, Speech and Dialogue*, pages 3–13. Springer.

Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C Courville. 2017. Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*, pages 3288–3294.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562. ACM.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Alan M. Turing. 1950. I.computing machinery and intelligence. *Mind*, LIX(236):433–460.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proceedings of the 31st International Conference on Machine Learning*.

Harm de Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. 2018. Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer.