Scalable Schema-Guided Dialogue State Tracking

Abhinav Rastogi, Amir Fayazi, Raghav Gupta, Ulrich Rueckert, Jindong Chen
Google AI, 1600 Amphitheatre Parkway, Mountain View, CA
{abhirast,amiraf,raghavgupta,rueckert,jdchen}@google.com

DSTC8 Track Proposal

1 Motivation

Virtual assistants such as the Google Assistant, Alexa, Siri, Cortana etc. help users accomplish tasks by providing a natural language interface to service providers (backends/APIs). Such assistants often support a large universe of third-party APIs to accomplish various tasks. Dialogue State Tracking (DST) is a core component of such assistants. It outputs the dialogue state after each user utterance, which is a summary of the entire conversation till the current turn. The dialogue state is used to invoke the APIs with appropriate parameters as specified by the user.

Deep learning-based approaches [1, 2, 3, 4] have achieved success in dialogue state tracking. Common public datasets for DST include DSTC2 [5], MultiWOZ [6] and M2M [7], all of which operate over a fixed set of slots across all examples. As a consequence, state of the art approaches on these datasets learn to recognize patterns in the dialogue without understanding the slots' semantics, limiting the capability to scale to unseen slot types and APIs. To this end, we propose the **scalable schema-guided dialogue state tracking** track. Our goal is to highlight the DST problem on unseen APIs given a schema of these target APIs, while supporting realistically many heterogeneous APIs with possibly overlapping functions. Zero-shot models utilizing domain and/or slot descriptions have been gaining popularity for spoken language understanding tasks [8, 9, 10]. This dataset aims to motivate similar approaches for dialogue state tracking.

2 Problem Setup

Each example in our dataset consists of a dialogue between a human and a virtual assistant along with schemas for one or more APIs relevant to the dialogue. The representation of the schema for an API is discussed in Section 2.1. The dialogue state labels after each user utterance in the dialogue are also provided. The dialogue state representation is conditioned on the schemas provided with the dialogue. It is expressed as an assignment of a single value for each slot for every schema present in the example. Apart from the values listed in the schema, two special values `null` (slot has not been specified yet) and `dontcare` (user has no preference for the slot) are also allowed. We propose multiple tasks for dialogue state tracking which are described in Section 2.2.

2.1 Schema Representation

We generate schemas corresponding to different tasks handled by virtual assistants (such as playing music, buying movie tickets, searching for restaurants). A schema defines the interface for a backend API. It contains a description and a set of slots. The description is a natural language summary of the function of the API. Slots in the schema correspond to API parameters. To help generalize to unseen slots, our schema representation also includes a natural language description of each slot. Slots are categorized into one of the following two types:

1. *Categorical*: A slot taking one of a finite set of possible values. The schema also includes the set of all possible values for such slots.

2. *Free-form*: Such slots can take any string value. In this dataset we will restrict these values to be derived from the conversation history. Otherwise it is not feasible to construct a fixed vocabulary for such values because of presence of unseen values in evaluation set. A few example values taken by these slots may also be provided in the schema.

Since we also focus on handling a large number of APIs, we generate a 'universe' of around 50-100 APIs spanning multiple domains, with some of these APIs having overlapping functionality. Furthermore, to ensure generalization to unseen APIs, a different, non-overlapping universe of APIs will be defined for the evaluation sets.

2.2 Dialogue State Tracking Tasks

We propose three dialogue state tracking tasks with progressively increasing complexity.

Task 1 - Single API: This task evaluates dialogue state tracking when the dialogue is restricted to a single API. Each dialogue is accompanied by a single schema for the corresponding API.

Task 2 - Multiple APIs: This task evaluates dialogue state tracking in situations which require invoking multiple APIs. Each dialogue is accompanied by up to 3 schemas. This task offers the challenges of disambiguating slots with similar functions across different APIs (e.g. *time* or *number_of_people*) and avoiding wrong predictions for APIs which have not been mentioned in the dialogue yet.

Task 3 - Multiple API Selection: This is an extension of task 2 with the set of API schemas pertaining to the dialogue are not known a priori. The model must select the relevant APIs from the universe of APIs as they become relevant to the dialogue. The dialogue state is represented over the selected APIs.

3 Dataset

We aim to create a universe of 50-100 APIs with possibly overlapping functionality from real-world sources spanning multiple domains covered by websites (e.g. flight bookings) and/or voice assistants (e.g. making reservations). With these APIs in place, we will simulate partially filled forms and collect dialogues geared towards the user specifying this set of values of API parameters to the assistant. We will collect dialogues using two approaches: the Wizard-of-Oz approach by pairing two crowdworkers to play the roles of the assistant and the user [6], and by developing agenda-based simulators for the user as well as the assistant [7]. A similar procedure will be repeated with a non-overlapping universe of APIs for obtaining the validation and test sets.

It is crucial that the API schemas in the validation and test sets were not seen during training. This will effectively evaluate zero-shot performance. Evaluating on unseen in-domain schemas (for instance, training on Uber and evaluating on Lyft) and on out-of-domain schemas are two manifestations of this challenge we will ensure are represented in the validation and test sets.

4 Evaluation

We will use the following evaluation metrics to evaluate the different submissions. A script for calculating these metrics will be provided with the dataset.

1. *Joint Goal Accuracy*: This is the strictest evaluation metric which corresponds to the fraction of user utterances for which the predicted dialogue state is exactly equal to the ground truth.
2. *Slot Precision/Recall/F1*: These metrics correspond to the predicted labels for individual slots in the dialogue state, microaveraged over all slots.
3. *Mean Average Precision for API retrieval*: Specifically for Task 3, this metric will evaluate the models' performance on retrieving the correct API(s).

Supplemental Material

API Schemas

Fandango: Movie Tickets & Movie Times			
Slot name	Datatype	Description	Example values
movie_name	*free-form*	Title of movie	-
theatre_name	*free-form*	Name or address of theatre	*Century 20, CineArts...*
date	*free-form*	Date of movie show	*March 24, tomorrow...*
time	*free-form*	Time of movie show	*8:20pm, noon...*
num_tickets	*category*	Number of tickets to buy	*1, 2...10*

Uber: Get a Ride Now			
Slot name	Datatype	Description	Example values
destination	*free-form*	Address of dropoff location	-
time	*free-form*	Time of pickup	*6pm, half past six...*
ride_type	*category*	Type of ride	*Express Pool, Pool, X...*
passengers	*category*	Number of passengers (for Pool/Express Pool)	*1, 2*

Utterances

Dialogue State

User: Find movie tickets for Aquaman tonight.

Fandango
movie_name: Aquaman
date: tonight

Assistant: Aquaman is playing at Century 16 at 7pm, 8:30pm and 9:40pm.

User: Can you look for tickets in AMC Mercado instead?

Fandango
movie_name: Aquaman
date: tonight
theatre_name: AMC Mercado

Assistant: Sorry, Aquaman is not playing there.

User: Okay, in that case, book 2 tickets for the 7pm show.

Fandango
movie_name: Aquaman
theatre_name: Century 16
date: tonight
time: 7pm
num_tickets: 2

Assistant: Your tickets were booked. Anything else I can help with?

User: Can you also book an Uber to get there?

Fandango: Same as previous
Uber
destination: Century 16

Assistant: Scheduled an Uber leaving at 6:30pm. Does that work?

User: Yes, perfect. That'll be all.

Fandango: Same as previous
Uber
destination: Century 16
time: 6:30pm

Figure 1: An example dialogue for Task 2, with schemas provided for two intents. The inputs are the API schemas (yellow) and the user and assistant utterances (red and green respectively). The expected output at each turn is the dialogue state as shown on the right.

References

[1] Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*, 2016.

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

[3] Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. Multi-task learning for joint language understanding and dialogue state tracking. *arXiv preprint arXiv:1811.05408*, 2018.

[4] Victor Zhong, Caiming Xiong, and Richard Socher. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1458–1467, 2018.

[5] Matthew Henderson, Blaise Thomson, and Jason D Williams. 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, 2014.

[6] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.

[7] Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*, 2018.

[8] Ankur Bapna, Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck. Towards zero-shot frame semantic parsing for domain scaling. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 2476–2480, 2017.

[9] Anjishnu Kumar, Pavankumar Reddy Muddireddy, Markus Dreyer, and Björn Hoffmeister. Zero-shot learning across heterogeneous overlapping domains. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017.

[10] Sungjin Lee and Rahul Jha. Zero-shot adaptive transfer for conversational language understanding. *arXiv preprint arXiv:1808.10059*, 2018.