DFEE: Interactive DataFlow Execution and Evaluation Kit

Han He†, 1, Song Feng†, 2, Daniele Bonadiman2, Yi Zhang2, Saab Mansour2

1 Emory University
2 AWS AI Labs
han.he@emory.edu, {sofeng, dbonadim, yizhngn, saabm}@amazon.com

Abstract
DataFlow has been emerging as a new paradigm for building task-oriented chatbots due to its expressive semantic representations of the dialogue tasks. Despite the availability of a large dataset SMCalFlow and a simplified syntax, the development and evaluation of DataFlow-based chatbots remain challenging due to the system complexity and the lack of downstream toolchains. In this demonstration, we present DFEE, an interactive DataFlow Execution and Evaluation toolkit that supports execution, visualization and benchmarking of semantic parsers given dialogue input and backend database. We demonstrate the system via a complex dialog task: event scheduling that involves temporal reasoning. It also supports diagnosing the parsing results via a friendly interface that allows developers to examine dynamic DataFlow and the corresponding execution results. To illustrate how to benchmark SoTA models, we propose a novel benchmark that covers more sophisticated event scheduling scenarios and a new metric on task success evaluation. The codes of DFEE have been released on https://github.com/amazon-science/dataflow-evaluation-toolkit.

Introduction
The representation of dialogue states and agent actions plays a key role in task-oriented dialogue systems as it defines user requests and agent behaviors. Popular representations include intent slots (Pieraccini et al. 1992), continuous representations (Bordes, Boureau, and Weston 2017), program synthesis (Gulwani, Polozov, and Singh 2017) and Dialog2API (Chen et al. 2020). Fixed symbolic representations are more trainable while distributed continuous representations are more extendable. As a structured yet flexible representation, DataFlow composes functions and data into a computational graph that interactively gets generated, revised and executed (Andreas et al. 2020). It has been shown to outperform traditional methods on compositional actions, complex anaphora and exception handling. Though lots of prominent features complemented DataFlow, they inevitably increase the complexity of downstream implementations. Recently, (Meron 2022) proposes OpenDF to simplify SMCalFlow (Andreas et al. 2020) annotations for a more readable syntax. However, developers still cannot directly use OpenDF to generate executable Python programs for a downstream task due to lack of a complete toolchain.

We present the DFEE toolkit, an interactive system that supports execution, evaluation and visualization of DataFlow graphs for event scheduling tasks. Our system allows developers to interact using natural language utterances and to inspect the executed DataFlow graphs along with responses in real time. We demonstrate our system based on a real-world application of DataFlow-based chatbot, virtual assistant for event scheduling that requires temporal reasoning. To better assess the DataFlow, we enhance the evaluation by supporting a new metric and a novel benchmark dataset that is challenging to SoTA models.

Our contributions are therefore three-fold: (1) we release a live DataFlow chatbot that allows researchers and practitioners to compile human language into DataFlow graphs and observe realtime responses; (2) we also propose and support execution accuracy as a metric on DataFlow for end-to-end evaluation of semantic parsers; (3) we demonstrate our benchmarking toolkit on a task and a new dataset that challenge the SoTA models.

Figure 1: DFEE modules: ① Chat, ② Dialog2API, ③ API Execution, ④ Exact Match, ⑤ Execution Accuracy.
Evaluation

Prior work adopts Exact Match (EM) as the sole metric to evaluate DataFlow programs (Roy et al. 2022). However, we observe that the same semantics could be represented with different programs due to API implementations or execution processes. For instance, a simple temporal phrase “10 at night” could be parsed to 3 different APIs from SMCalFlow: NumberPM(10), HourMilitary(22) or HourMinutePm(hour=10, minute=0). Thus, we propose a new metric and a dataset to demonstrate the necessity of evaluating end-to-end execution results via our system.

Metrics

To better assess the DataFlow programs, we propose to support a new metric - Execution Accuracy (EA). It is defined as the portion of turns correctly executed. A correctly executed turn has to make the agent generate the same response as the oracle response and ensure the backend database matches the oracle one.

Benchmark

To support a more complete evaluation of both EM and EA on the exemplar task, a dataset consisting of 1000 dialogues is created by appending temporal reasoning utterances to SMCalFlow dialogues. It provides the annotations on the DataFlow graphs and the expected execution results based on a database as an oracle. Specifically for the dataset in the calendar domain, we design a set of calendar events before and after the execution per each turn.

Model

We train and evaluate seq2seq (Roy et al. 2022) and transition parsers (Zhou et al. 2022) on the BenchCLAMP (Roy et al. 2022) split of SMCalFlow. Both kinds of parsers take as input the last turn and the current user utterance. Another BART model (BART-Ours) is also trained on OpenDF simplified version to evaluate the impact of program syntax.

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall</th>
<th>Non-Tmp</th>
<th>Tmp</th>
<th>Sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5</td>
<td>80.2</td>
<td>80.4</td>
<td>53.3</td>
<td>27.2</td>
</tr>
<tr>
<td>BART</td>
<td>81.0</td>
<td>81.2</td>
<td><strong>60.0</strong></td>
<td>1.7</td>
</tr>
<tr>
<td>RoBerta</td>
<td>76.5</td>
<td>76.6</td>
<td>53.8</td>
<td><strong>0.1</strong></td>
</tr>
<tr>
<td>BART-Ours</td>
<td><strong>82.9</strong></td>
<td><strong>83.1</strong></td>
<td>53.3</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Table 1: EM and latency in seconds (sec) on BenchCLAMP testset, non-temporal (Non-Tmp) and temporal (Tmp) turns. † and ‡ indicate our experiments of seq2seq (Roy et al. 2022) and transition (Zhou et al. 2022) models respectively. All transformers are large versions.

As shown in Table 1, BART-Ours outperforms BART by 1.9, aligning with the finding by Meron (2022). The transition based parser RoBerta is optimized for latency but not accuracy. So, we opt for BART-Ours for the later experiments and use it as the default parser in the DFEE system.

System

We modularize the system into three components: semantic parser, execution engine, and user interface. These components are connected via HTTP requests such that they are swappable. This decoupled design enables the developers to plug-and-play new parser components for benchmarking regardless of the model or programming language.

We build our execution engine upon the OpenDF framework and enhance it in two ways to significantly improve its usability: (1) adding the fulfillment of the necessary APIs for event scheduling tasks; (2) adding worker processes to support multiple concurrent access to the system.

Our system provides two interfaces: (1) one is the conversational GUI with the visualization of parsing results, i.e., converting dialogue input to an executable program (APIs); (2) one is for CLI on benchmarking our proposed Execution Accuracy metric for SMCalFlow.

GUI for Dialog2API

As illustrated in Figure 1, the flow of our GUI is outlined below: one enters an utterance in the chat; the utterance is parsed into a piece of DataFlow code (Dialog2API); the code is executed and the resulting DataFlow graph is visualized; the updates to user’s database are reflected to the calendar; the agent response generated during execution is displayed on the chat.

CLI for evaluation

We provide a CLI for benchmarking EM and end-to-end EA of a model. To benchmark a new model, the developers can either provide an online RESTful API URL, or offline parsed DataFlow expressions. The expressions (either parsed or provided) will be executed on our system and the execution and evaluation results will be returned. To demonstrate its functionality, we benchmark BART-Ours on SMCalFlow and our own dataset in Table 2.

The model performed significantly worse on the temporal turns, especially on our challenging benchmark that contains diverse scenarios and conversational language styles, appealing for future attention to temporal reasoning and robustness. EA for each operation is generally higher than EM accuracy, due to the fact that there are many possible ways to express the same semantics, which indicates that EM might be insufficient for evaluating DataFlow programs.

Conclusion

We presented a novel DataFlow execution and evaluation kit, DFEE. This interactive system has shown to be user-friendly and extendable due to its modularized structure. We further proposed Execution Accuracy metric for SMCalFlow and benchmarked a SoTA parser on a richly annotated dataset. We hope our system can facilitate the future development of robust semantic parsers and intelligent dialogue systems.
References


