Workflow **D**iscovery from Dialogues in the Low Data Regime

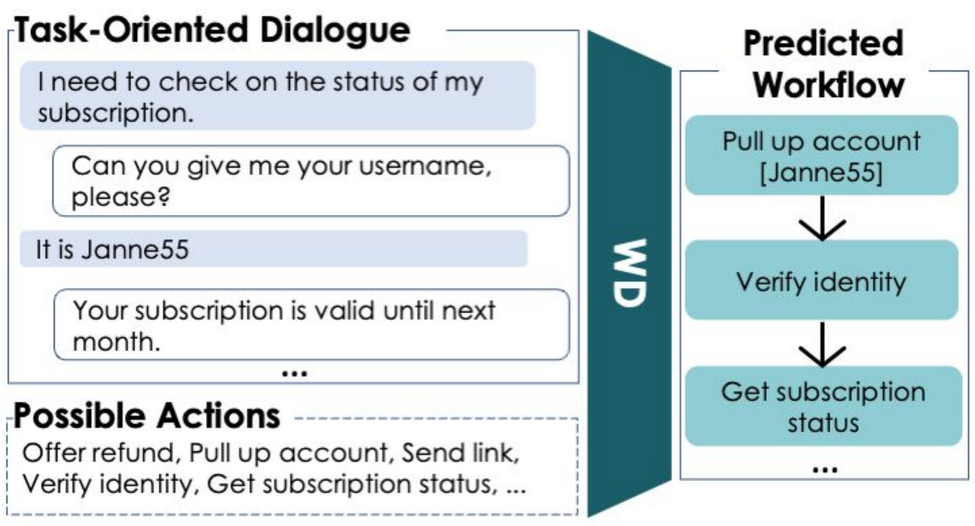
Amine El Hattami[*123], Issam Laradji[1], Stefania Raimondo[1], Pau Rodríguez[1], David Vazquez[1], Chris Pal[123]
[1]Polytechnique Montreal, [2]Mila, [3]ServiceNow Research, [*]amine.elhattami@servicenow.com

Workflow Discovery (WD)

Workflow Discovery (WD) is targeted toward **extracting workflows from dialogues between real people**. The extracted workflow is a sequence of actions with their respective slot values in the same order in which they appeared during the conversation.

The extracted workflows can be used to:

- Train human agents
- Help transition to automated dialogue systems
- Guide analysts to understand if an unresolved problem is due to a divergence from a formal workflow or if workflows have organically emerged



Given a dialogue and an optional list of possible actions, a model is expected to predict the target workflow.

Why WD ?

Task-oriented dialogues are common in everyday life to assist users with various activities such as online shopping or solving complex problems. Behind these dialogues, there are often implicit or explicit workflows. For example, booking an airline ticket might involve the workflow: (1) pull up an account, (2) register a seat, and (3) request payment.

Services **without formal workflows** can struggle to manage **variations in problem resolution**, especially when agents follow different "unwritten rules". This variation can **affect negatively user satisfaction** and make training new agents challenging. Further, when transitioning from human to virtual agents, identifying these workflows can require significant manual effort and domain expertise, particularly when possible actions and procedures evolve over time (e.g., return policy changes). Thus, **extracting workflows from dialogues automatically and at scale can bring significant value for both human and virtual agent systems**.

Contributions

- Novel task formulation to the problem of workflow extraction from dialogues, we call **Workflow Discovery (WD)**.

- **WD Datasets** created from the Action Based Conversations Dataset (ABCD) and MutliWOZ dataset.

- **Multiple WD baselines** using a text-to-text approach. We test the performance in multiple learning settings: (1) In-Domain, (2) Cross-Domain Zero-Shot, (3) Cross-Dataset Zero-Shot, and (4) Cross-Dataset Few-Shot.

- A conditioning mechanism for our text-to-text approach, providing the model with a set of possible actions to use as potential candidates and show its efficacy on improving the adaptation performance.

- To ensure strong baselines, we tested our text-to-text approach, and show that it achieves state-of-the-art results on related but different dialogue tasks with existing baselines (i.e., Action State Tracking (AST) and Cascading Dialogue Success (CDS)).

WD Vs. Existing Tasks

- WD focuses on extracting the sequence of actions taken to resolve an issue, which may not be known beforehand and have to be created by the WD model.

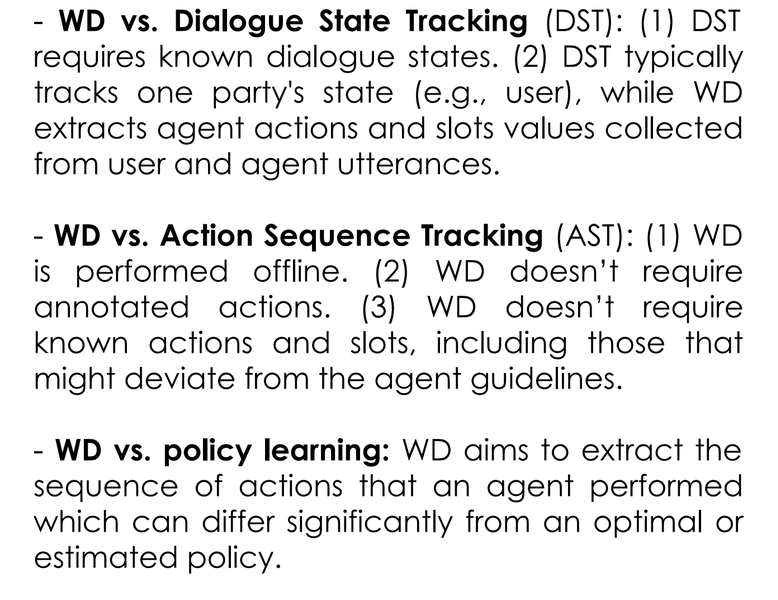
- **WD vs. Dialogue State Tracking** (DST): (1) DST requires known dialogue states. (2) DST typically tracks one party's state (e.g., user), while WD extracts agent actions and slots values collected from user and agent utterances.

- **WD vs. Action Sequence Tracking** (AST): (1) WD is performed offline. (2) WD doesn't require annotated actions. (3) WD doesn't require known actions and slots, including those that might deviate from the agent guidelines.

- **WD vs. policy learning:** WD aims to extract the sequence of actions that an agent performed which can differ significantly from an optimal or estimated policy.

- **WD vs. open world intent and slot induction**: WD focuses on extracting actions taken by an agent in the right order as opposed to standalone user intents.

Generalization to New Domains



A Model Trained on WD is expected to be able to **formulate new compound keywords** to characterize new actions as well as extract their slot values for actions that are not a part of the known action domain.

Baselines

Evaluation

- We evaluate the WD task using the Exact Match (EM) and Cascading Evaluation (CE).

- We use BERTScore for all zero-shot experiments. We assume that a predicted action is valid if it has an F1-score above 95% (e.g., "check customer identity" ~ "verify customer identity").

In-Domain Results

Models	EM/CE	
	WITHOUT POSSIBLE ACTIONS	WITH POSSIBLE ACTIONS
T5-Small (60M)	44.1/67.9	44.8/68.6
T5-Base (220M)	47.7/69.9	49.5/72.3
BART-Large (406M)	42.0/60.3	44.9/64.3
PEGASUS-Large (568M)	49.9/71.2	52.1/72.6
T5-Large (770M)	50.6/73.1	**55.7/75.8**

- the EM and CE scores show an expected improvement as we scale the model size.

- BART-Large variant that showed an interesting behavior where it struggles with slot values representing identifiers (e.g., Amine34).

- Predicting slot values is much more challenging and most performance improvements as we increase in the model size can be attributed to an enhanced slot value extraction capability.

Cross-Domain Zero-Shot Results

Models	Promo Code EM/CE	
	WITHOUT POSSIBLE ACTIONS	WITH POSSIBLE ACTIONS
T5-Small (60M)	42.3/65.1	42.5/66.5
T5-Base (220M)	46.0/67.6	47.8/69.7
BART-Large (406M)	41.5/58.3	43.6/62.2
PEGASUS-Large (568M)	47.4/68.8	49.6/69.2
T5-Large (770M)	48.1/70.7	**51.8/72.3**

- We used a "leave-one-out" setup.

- Models confuse similar actions (e.g., "offer a refund" and "offer promo code").

- Generating actions from the omitted domain is challenging for smaller models but adding the possible actions helps.

Cross-Dataset Zero-Shot Results

Models	EM/CE	
	WITHOUT POSSIBLE ACTIONS	WITH POSSIBLE ACTIONS
T5-Small (60M)	0.0/0.0	5.3/8.4
T5-Base (220M)	3.6/10.0	23.0/38.3
BART-Large (406M)	5.1/12.2	24.1/39.9
PEGASUS-Large (568M)	9.8/15.2	27.4/41.2
T5-Large (770M)	9.0/13.1	**26.9/40.0**

- All model variants performed poorly when the list of actions was not included in the input.

- Adding possible actions helps, but performance remains low compared to in-domain results.

Cross-Dataset Few-Shot Results

k	# Samples	EM/CE	
		WITHOUT POSSIBLE ACTIONS	WITH POSSIBLE ACTIONS
1	11	5.9/54.8	8.2/58.7
5	55	24.4/65.4	61.7/84.4
10	106	43.2/73.0	**72.2/89.1**

- All models shows a significant adaptation performance that keeps increasing as training samples increase.

- Adding possible actions helps, but performance remains low compared to in-domain results.

- Most failures are due to invalid slot values prediction.

Resources

Public Repository & Dataset

Papers with Code Leaderboard

ICML International Conference On Machine Learning