# N-Shot Learning for Augmenting Task-Oriented Dialogue State Tracking

***Taha Aksu, Zhengyuan Liu, Min-Yen Kan, Nancy F. Chen***

## Introduction

- Few-shot domain adaptation of DST models is a crucial problem because it is common that new business models prefer dialogue agents over static websites to serve customers.
- We propose a dataset-level augmentation for few-shot domain adaptation, unlike previous datum level methods.

## Intuition

- We use belief states of task-oriented dialogues as blueprints and mingle turns between dialogues to construct new synthetic dialogues.

## Methodology

1. Assign a dialogue function to each turn pair in the dialogue.
2. Break down each dialogue pair by pair into pieces storing with their dialogue functions.
3. Combine these pieces generating new dialogues while making sure consecutive pairs have complementary functions.

## Experiments

- We use 5 domains from the MultiWOZ dataset.
- At each iteration we leave one domain out and:
  - Train on the other four domains.
  - Finetune and test on the left-out domain.
- We repeat experiments with both original and augmented samples.
- We use TRADE and TOD-BERT models for experiments.

# Augmenting Task-Oriented Dialogues at the **dataset-level** outperforms some **learning-based** methods on few-shot domain adaptation.

**<u>Original Dialogue</u>**

U → Hi , I am looking for a train that is going to **Cambridge** and arriving there by **20:45**, is there anything like that?

A → There are many trains like that. Where will you be departing from?

U → I am departing from **Birmingham New Street**.

A → Can you confirm your desired travel day?

U → I would like to leave on **Wednesday**.

A → Okay, we have a ticket that is fit, should I book it?

U → Yes, please.

**<u>Turn-pair template</u>**

**De-lexicalized Turns:**

A → There are many trains like that. Where will you be departing from?

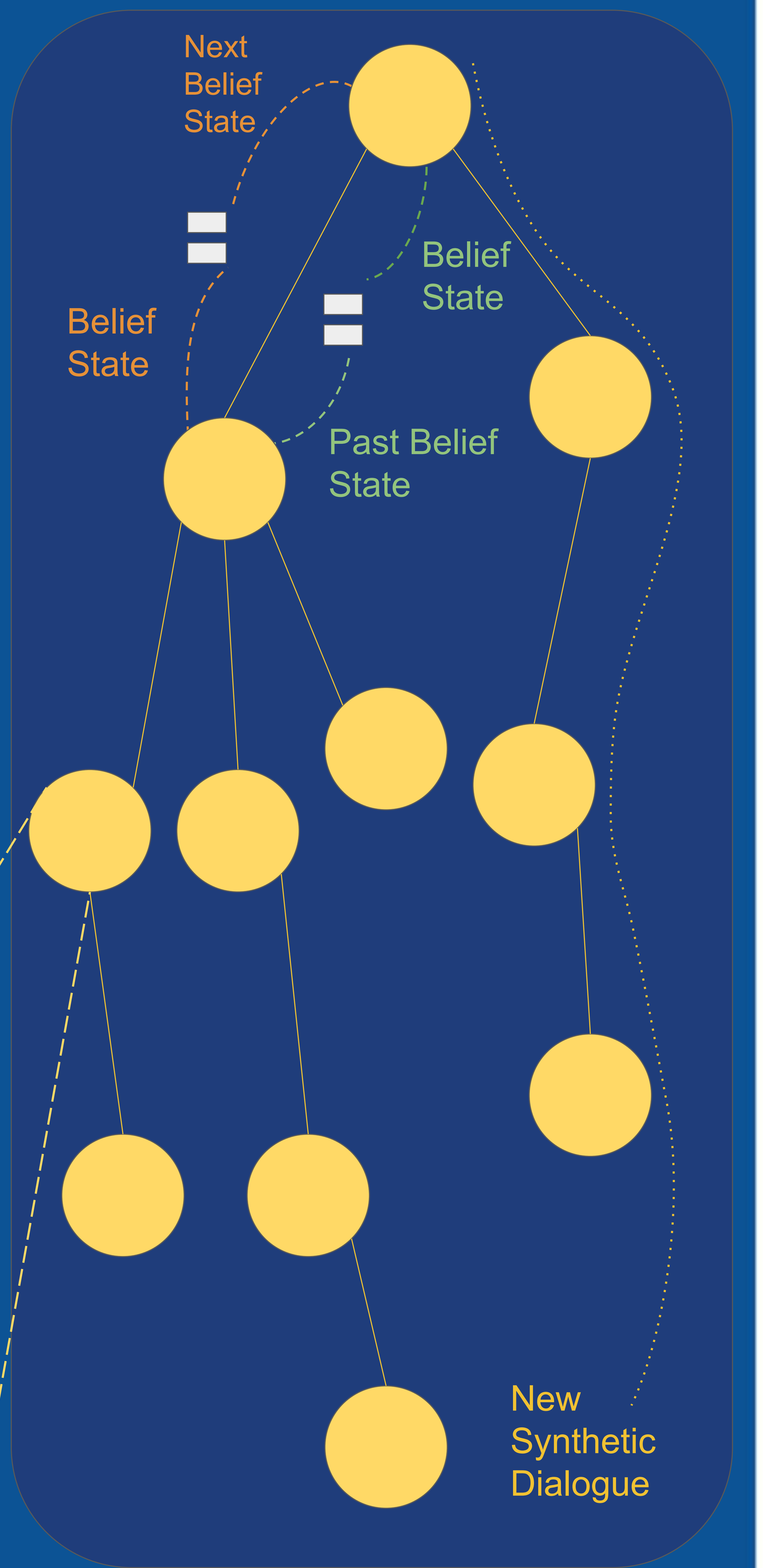
U → I am departing from **[train-departure]**.

**<u>Turn-pair Function</u>**

**BS:** {train-dest, train-arv_time, train-dep}

**Past BS:** {train-dest, train-arv_time}

**Next BS:**{train-dest, train-arv_time, train-dep, train-day}




# Main Findings

→ Augmenting TODs on a **<u>dataset</u>** level rather than on a **<u>datum level</u>** harbors better performance for n-shot fine tuning.

→ Exploiting the **<u>organized structure</u>** in a TOD's belief state is an effective way to **<u>assign functions to turns</u>** and thus **<u>break down</u>** dialogues into smaller pieces.

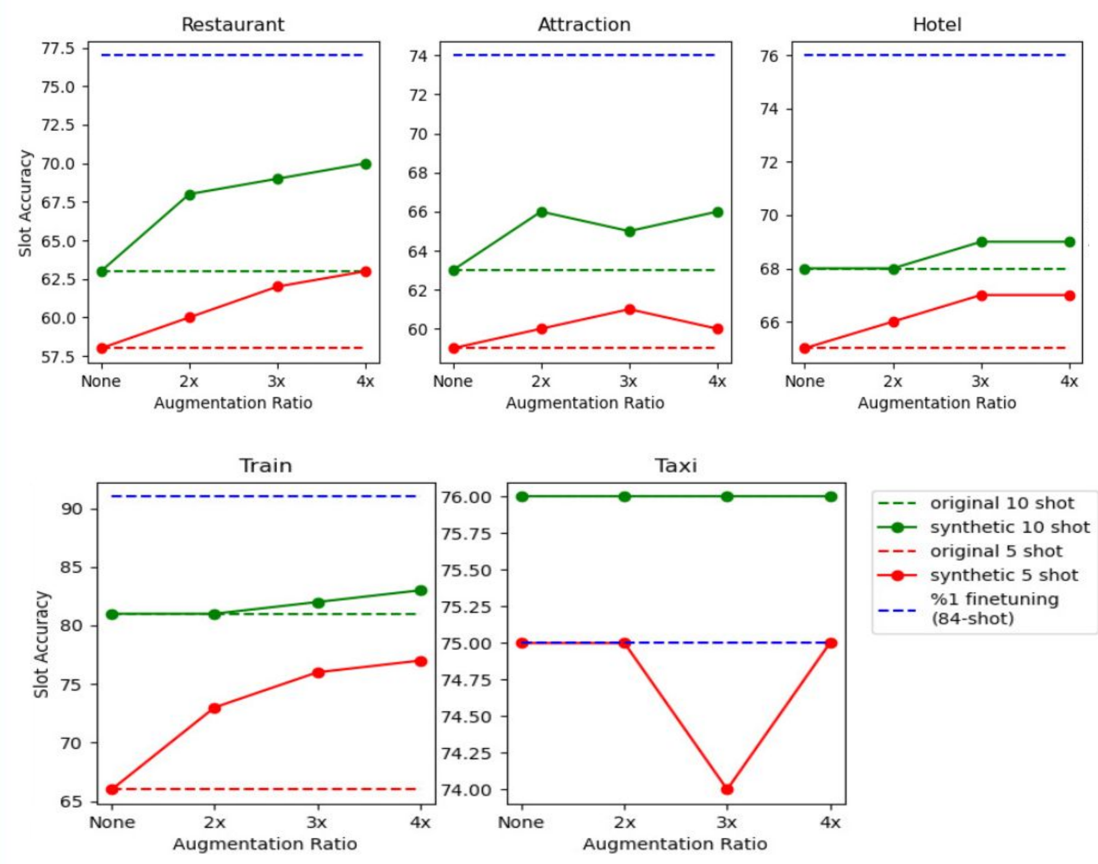
→ **<u>Augmentation</u>** is only one way to utilize this break-down and we hope to see further studies that apply it to other aspects of TODs such as **<u>intent recognition</u>**, **<u>response generation</u>**, etc.

## TRADE Few-shot Experiments

| | Rest. | | Attr. | | Train | |
|---|---|---|---|---|---|---|
| | JA | SA | JA | SA | JA | SA |
| BM | 0.12 | 0.54 | 0.18 | 0.54 | 0.22 | 0.49 |
| BM+ | 0.21 | 0.77 | 0.43 | 0.74 | 0.61 | 0.91 |
| 5S | | | | | | |
| Orig. | 0.12 | 0.58 | 0.25 | 0.59 | 0.25 | 0.66 |
| CoCo | **0.13** | **0.62** | 0.24 | 0.58 | 0.27 | 0.69 |
| Ours | **0.13** | **0.62*** | **0.26** | **0.61** | **0.31** | **0.77*** |
| 10S | | | | | | |
| Orig. | 0.13 | 0.63 | 0.30 | 0.63 | 0.37 | 0.81 |
| CoCo | **0.16** | 0.67 | 0.31 | 0.64 | **0.39** | 0.82 |
| Ours | **0.16*** | **0.70*** | **0.32*** | **0.66*** | **0.39** | **0.83** |

→ First two rows: zero shot (**lower bound**), 1% fine tuning (**upper bound**). Consecutive sections show 5 and 10 shot fine tuning with **original shots**, **CoCo** augmentation and **our augmentation** respectively.

## Effect of Augmentation Ratio




→ Our framework **outperforms** base fine-tuning **steadily**, and the amount of synthetic data affects the results **proportionally**.

## How Does Augmentation Improve Performance?

| Active Slot F1 | Unseen Values | Seen Values |
|---|---|---|
| All | | |
| Orig. | 0.1 e-3 | 0.24 |
| Aug. | **0.2 e-3** | **0.28** |
| Rest. | | |
| Orig. | 1.5 e-3 | 0.20 |
| Aug. | **2.3 e-3** | **0.26** |

→ Our framework helps to exploit slots that have a **bounded value pool** with less unique values and also slots with **more frequent unseen values**.

| Active Slot F1 | Rest. | Hotel |
|---|---|---|
| 5S | | |
| Full | **0.183** | **0.255** |
| SR | 0.157 | 0.250 |
| 10S | | |
| Full | 0.198 | **0.258** |
| Sr | **0.237** | 0.243 |

→ Template generation improves results compared against **only surface realization** in most cases.

**Scan the QR code to view all figures/tables and the paper.**

*Connect with the first author!*

taksu@u.nus.edu