Zero-Shot Transfer Learning with Synthesized Data for Multi-Domain Dialogue State Tracking

Giovanni Campagna Agata Foryciarz Mehrad Moradshahi Monica S. Lam
Computer Science Department
Stanford University
Stanford, CA, USA
{gcampagn,agataf,mehrad,lam}@cs.stanford.edu

Abstract

Zero-shot transfer learning for multi-domain dialogue state tracking can allow us to handle new domains without incurring the high cost of data acquisition. This paper proposes new zero-short transfer learning technique for dialogue state tracking where the in-domain training data are all synthesized from an abstract dialogue model and the ontology of the domain. We show that data augmentation through synthesized data can improve the accuracy of zero-shot learning for both the TRADE model and the BERT-based SUMBT model on the MultiWOZ 2.1 dataset. We show training with only synthesized in-domain data on the SUMBT model can reach about 2/3 of the accuracy obtained with the full training dataset. We improve the zero-shot learning state of the art on average across domains by 21%.

1 Introduction

Automated conversational agents can reduce the costs of customer support, a necessary service in just about every business. However, training a goal-directed dialogue agent for a domain often requires acquiring annotated dialogues to cover all possible conversation flows. Commonly, this is done using the Wizard-of-Oz technique (Kelley, 1984), where two crowdsource workers converse with each other, while also annotating the state at each turn. This technique has been employed to construct several datasets (Hemphill et al., 1990; Wen et al., 2017; Yu et al., 2019). Recently, it has been used to build the MultiWOZ dataset (Budzianowski et al., 2018), a large corpus of dialogues across 7 domains.

Unfortunately, not only is the initial acquisition expensive, annotating dialogues correctly has proven to be challenging due to human errors, delays in annotation, inconsistent conventions, and normalization issues (Eric et al., 2019; Zhou and Small, 2019). The MultiWOZ dataset still has significant inconsistencies (Zhou and Small, 2019) despite having been constructed through multiple rounds of annotations (Budzianowski et al., 2018; Eric et al., 2019).

S: START
U: Can you help with information regarding a food place? I need to book at 15:45.
S: SEARCHREQUEST restaurant(book_time = "15:45")
A: How about the restaurant with name La Tasca and Italian food?
U: Can you find something which serves seafood?
S: SEARCHREQUEST restaurant(book_time = "15:45", food = "seafood")
A: What date are you looking for?
U: Thursday please.
S: SEARCHREQUEST restaurant(book_time = "15:45", food = "seafood", book_day = "thursday")
A: How about the Copper Kettle? It is a food place with seafood food.
U: What is the price range and the area?
S: SLOTQUESTION restaurant(book_time = "15:45", food = "seafood", book_day = "thursday", price_range =?, area =?)
A: The Copper Kettle is a moderately priced restaurant in the north of the city. Would you like a reservation?
U: No, thanks.
S: CLOSE restaurant(book_time = "15:45", food = "seafood", book_day = "thursday")
A: Can I help with you anything else?
U: Thank you, that will be it for now.
S: END restaurant(book_time = "15:45", food = "seafood", book_day = "thursday")

Figure 1: An example of a dialogue that can be synthesized from our templates. 'U:' indicates the user, 'A:' the agent, and 'S:' is the dialogue state at each turn.

We observe empirically from the MultiWOZ training data that conversations in all the domains follow the same pattern: the agent and user start by greeting each other, tehn they converse to find a proposal that satisfies the user, the user provides additional required information, and finally the agent completes the user's transaction.

To facilitate transfer learning, we create an abstract model of dialogues that is independent of the domain of the conversation. In this paper we will

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 122–132
July 5 - 10, 2020.

focus on dialogues for transactions; other kinds of dialogues such as opinion sharing will have different models. We have developed an algorithm that accepts an ontology of a domain and a few phrases commonly used in that domain. The algorithm synthesizes dialogue training data based on an abstract dialogue model. The dialogue synthesized consists of turns of conversation, each of which has a start state, an agent utterance, a user utterance, and an end state. The start and end states summarize the semantics of the conversation at those points. An example of a dialogue that can be synthesized by our model is shown in Fig. 1.

To transfer knowledge to a new domain in a zero-shot setting, we train with the synthesized data for the new domain together with existing data for other domains. In addition, we adapt training samples from related domains by substituting them with the vocabulary of the new domain. We can improve the accuracy of the abstract dialogue model as well as the state-tracking neural network by iteratively refining the model based on the error analysis on the validation data, and by introducing additional annotations in the new domain. Note that the abstract dialogue model can be also used directly to implement the agent itself.

The contributions of this paper are as follows:

- A new zero-short transfer learning technique for dialogue state tracking where the in-domain training data are all synthesized from an abstract dialogue model and the ontology of the domain.
- Our approach improves over the previous state-of-the-art result on zero-shot transfer learning for MultiWOZ 2.1 tasks by 21% on average across domains.
- We show that our approach improves the accuracy for TRADE (Wu et al., 2019), an RNN-based model, and SUMBT (Lee et al., 2019), a BERT-based model (Devlin et al., 2019), suggesting that our technique is independent of the specific model used.
- Our experimental results show that synthesized data complements BERT pretraining. The BERT-based SUMBT model can, in a purely zero-shot fashion, achieve between 61% and 92% of the accuracy obtained by a model trained on the full dataset. We propose combining pretrained models with synthesized data as a general technique to bootstrap new dialogue state trackers.

2 Related Work

Dialogue Datasets and Synthesis. Synthesized data (in training and evaluation) was proposed by Weston et al. (2015) to evaluate the ability of neural models to reason compositionally, and was also used in visual question answering (Johnson et al., 2017a; Hudson and Manning, 2019) and semantic parsing (Lake and Baroni, 2018).

Wang et al. (2015) proposed synthesizing data, then crowdsourcing paraphrases to train semantic parsers. Various semantic parsing datasets have been generated with this technique (Su et al., 2017; Zhong et al., 2017) and the technique has also been adapted to the multiturn setting (Cheng et al., 2018; Shah et al., 2018). While it tends to be well-annotated, paraphrase data is expensive to acquire, and these datasets are very small.

More recently, we proposed training with both a large amount of synthesized data and a small amount of paraphrase data for semantic parsing of single sentences (Campagna et al., 2019; Xu et al., 2020). We showed that training with such data can perform well on real-world evaluations. This paper extends this work to the multi-turn setting. Dialogues are more complex as they need to capture information, such as the abstract dialogue state, that is not present in the target annotation (domain and slot values). We extend the synthesis algorithm to operate based on a dialogue model, tracking enough information to continue the dialogue. We also present a novel dialogue model that is suitable for synthesis.

Dialogue State Tracking. Dialogue state tracking is a long-studied field, starting with the first *Dialogue State Tracking Challenge* (Williams et al., 2014). A review of prior work can be found by Williams et al. (2016).

Previous works on DST use different approaches, ranging from using handcrafted features to elicit utterance information (Henderson et al., 2014; Wang and Lemon, 2013). Mrkšić et al. (2017) use Convolutional Neural Networks to learn utterance representations. However, their models do not scale as they do not share parameters across different slots. Zhong et al. (2018) and Nouri and Hosseini-Asl (2018) propose a new global module that shares information to facilitate knowledge transfer. However, they rely on a predefined ontology. Xu and Hu (2018) use a pointer network with a Seq2Seq architecture to

handle unseen slot values. Lee et al. (2019) use a pre-trained BERT model (Devlin et al., 2019) to encode slots and utterances and uses multi-head attention (Vaswani et al., 2017) to find relevant information in the dialogue context for predicting slot values. Wu et al. (2019) introduce an encoder-decoder architecture with a copy mechanism, sharing all model parameters between all domains. Zhou and Small (2019) formulate multi-domain DST as a question answering task and use reading comprehension techniques to generate the answers by either span or value prediction.

Johnson et al. (2017b) propose single encoder-decoder models for zero-shot machine translation by encoding language and input sentence jointly, and Zhao and Eskenazi (2018) propose cross-domain zero-shot language generation using a cross-domain embedding space.

Modelling of Dialogues. Previous work already proposed general models of dialogues as finite state machines (Jurafsky et al., 1997; Bunt et al., 2017; Yu and Yu, 2019). Existing models are optimized for analyzing existing human conversations. Our dialogue model is the first suitable for synthesis, carrying enough information to continue the dialogue.

Gupta et al. (2018) previously proposed a different annotation scheme for dialogues, using a hierarchical representation scheme, instead of the more typical intent and slot. Their work is complementary to ours: our method of dialogue synthesis is applicable to any annotation scheme. In this paper, we focus on the existing annotation scheme used by the MultiWOZ dataset.

3 Dialogue-Model Based Synthesis

In this section, we first define abstract dialogue models, then describe how we can generate dialogues based on the model. We also describe the techniques we use to adapt training dialogues from other domains to the new domain.

3.1 Abstract Dialogue Model

We define a *dialogue model* with finite sets of *abstract states*, *agent dialogue acts*, *user dialogue acts*, and *transitions*, defined below. The abstract dialogue for transactions we use in this paper is shown in Table 1.

The *abstract states* capture the typical flow of a conversation in that model, regardless of the domain. For example, a transaction dialogue model has states GREET, SEARCHREQUEST, COMPLETEREQUEST, COMPLETETRANSACTION, and CLOSECONVERSATION, etc. Each domain has a set of *slots*; each slot can be assigned a *value* of the right type, a special DONTCARE marker indicating that the user has no preference, or a special "?" marker indicating the user is requesting information about that slot. Thus, we can summarize the content discussed up to any point of a conversation with a *concrete state*, consisting of an abstract state, and all the slot-value pairs mentioned up to that point. Where it is not ambiguous, we refer to the concrete state as the *state* for simplicity.

All possible agent utterances in a dialogue model are classified into a finite set of *agent dialogue acts*, and similarly, all the possible user utterances into a finite set of *user dialogue acts*. Examples of the former are GREETUSER, ASKQUESTION, ANSWER, OFFERRESERVATION; examples of the latter are ASKBYNAME, ADDCONSTRAINTS, ACCEPT, REJECT.

Each *transition* in the model describes an allowed *turn* in a dialogue. A transition consists of an abstract start state, an agent dialogue act, a user dialogue act, and an abstract end state.

3.2 Dialogues from an Abstract Model

A *dialogue* is a sequence of turns, each of which consists of a start state, an agent utterance, a user utterance, and an end state. We say that a dialogue belongs to a model, if and only if,

1. for every turn, the start state's abstract state, the dialogue act of the agent utterance, the dialogue act of the user utterance, and the end state's abstract state constitute an allowed transition in the model.

2. the slot-value pairs of each end state are derived by applying the semantics of the agent and user utterances to the start state.

3. the first turn starts with the special START state, and every turn's end state is the start state of the next turn, except for the last turn, where the end state is the special END state.

3.3 Synthesizing a Turn with Templates

We use templates to synthesize dialogues in a domain from an abstract dialogue model and a domain ontology. In this paper, we introduce *dialogue model templates* which specify with grammar rules how to generate a turn of a dialogue from

From Abstract State	Agent Dialogue Act	User Dialogue Act	To Abstract State
Start		Greet	Greeting
		Ask by name	Info request
		Ask with constraints	Search request
Greet	Greet	Ask by name	Info request
		Ask with constraints	Search request
Search request	Ask to refine search	Provide constraints	Search request
	Ask question	Answer question	Search request
	Propose constraint	Accept constraint	Search request
		Add constraints	Search request
	Propose entity	Accept	Complete request
		Add constraints	Search request
		Reject	Search request
		Ask slot question	Slot question
		Ask info question	Info question
	Empty search, offer change	Change constraints	Search request
		Insist	Insist
Info request	Provide info, offer reservation	Accept	Accept
		Provide reservation info	Accept
		Ask info question	Info question
Info question	Answer, offer reservation	Accept	Accept
		Provide reservation info	Accept
		Thanks	Close conversation
Slot question	Answer, offer reservation	Accept	Accept
		Add constraint	Search request
Insist	Repeat empty search	Apologize	Close conversation
		Change constraints	Search request
Complete request	Offer reservation	Accept	Accept
		Thanks	Close conversation
Accept	Ask missing slots	Answer question	Complete transaction
Complete transaction	Execute	Ask transaction info	Transaction info question
		Thanks	Close conversation
	Error	Thanks	Close conversation
Transaction info question	Answer	Thanks	Close conversation
Close conversation	Anything else	Thanks	End

Table 1: Our abstract dialogue model for transaction dialogues. Each row represents one transition between abstract dialogue states.

a transition in the abstract model. They create possible agent and user utterances matching the agent and user dialogue acts in the transition, and they include a *semantic function* to ensure the utterances make sense given the input state. For example, the user should ask about a slot only if its value is not already known. The semantic function returns an output state that matches the semantics of the utterances. The slot values of the output state are used as the annotation for the turn when training the dialogue state tracker.

As an example, the SLOTQUESTION template shown in Fig. 2 corresponds to the 13th transition in the dialogue model in Table 1. The following agent and user utterances, separated by a delimiting token <sep>, are examples of dialogue acts PROPOSEENTITY and ASKSLOTQUESTION. They transition the abstract state SEARCHREQUEST to the abstract state SLOTQUESTION.

State: SEARCHREQUEST restaurant(...)
Agent: How about Curry Garden? It is an Indian restaurant in the south of town. <sep>
User: Is it expensive?
State: SLOTQUESTION restaurant(..., price = "?")

In this case, the non-terminals NAME, NP, ADJ_SLOT are expanded into domain-dependent phrases "Curry Garden", "Indian restaurant in the south of town", and "expensive", respectively, and the results of their semantic functions, *name, np, adj_slot*, are (sets of) slot-value pairs: name = "Curry Garden"; { food = "Indian", area= "south" }; price = "expensive". The semantic function of SLOTQUESTION checks that the input state does not already include a value for the price slot, and the price is not mentioned by the agent at this turn. It returns, as the new state, the old state with a "?" on the price.

All the non-dialogue specific templates are introduced by Xu et al. (2020). We have extended this template library, originally intended for database queries, to return slot-value pairs as semantic function results. Readers are referred to Xu et al. (2020) for details. This library has

```
SLOTQUESTION := "How about" NAME "? It is a " NP "."
                "<sep> Is it" ADJ_SLOT "?":
                λ(state, name, np, adj_slot) → {
                  if adj_slot ∈ (state.slots ∪ np)
                    return ⊥
                  state.abstract = SLOTQUESTION
                  state.slots[adj_slot.name] = "?"
                  return state
                }
NP := ADJ_SLOT NP : λ(adj_slot, np) → np ∪ {adj_slot}
NP := NP PREP_SLOT : λ(np, prep_slot) → np ∪ {prep_slot}
NP := "restaurant" : λ() → ∅

ADJ_SLOT := FOOD | PRICE : λ(x) → x
PREP_SLOT := "in the" AREA "of town" : λ(x) → x
NAME := "Curry Garden" | ... : λ(x) → name = x
FOOD := "Italian" | "Indian" | ... : λ(x) → food = x
AREA := "north" | "south" | ... : λ(x) → area = x
PRICE := "cheap" | "expensive" | ... : λ(x) → price = x
```

Figure 2: The SLOTQUESTION template and other non-dialogue specific templates used to generate the example interaction.

four kinds of domain templates. *Domain Subject Templates* describe different noun phrases for identifying the domain. *Slot Name Templates* describe ways to refer to a slot name without a value, such as "cuisine", "number of people" or "arrival time". *Slot Value Templates* describe phrases that refer to a slot and its value; they can be a noun phrase ("restaurants with Italian food"), passive verb phrase ("restaurants called Alimentum"), active verb phrase ("restaurants that serve Italian food"), adjective-phrase ("Italian restaurants"), preposition clauses ("reservations for 3 people"). Finally, *Information Utterance Templates* describe full sentences providing information, such as "I need free parking", or "I want to arrive in London at 17:00". These are domain-specific because they use a domain-specific construction ("free parking") or verb ("arrive").

Developers using our methodology are expected to provide domain templates, by deriving them manually from observations of a small number of in-domain human conversations, such as those used for the validation set.

3.4 Synthesizing a Dialogue

As there is an exponential number of possible dialogues, we generate dialogues with a randomized search algorithm. We sample all possible transitions uniformly to maximize variety and coverage. Our iterative algorithm maintains a fixed-size working set of incomplete dialogues and their current states, starting with the empty dialogue in the START state. At each turn, it computes a random sample of all possible transitions out of the abstract states in the working set. A fixed number of transitions are then chosen, their templates expanded and semantic functions invoked to produce the new concrete states. Extended dialogues become the working set for the next iteration; unextended ones are added to the set of generated results. The algorithm proceeds for a maximum number of turns or until the working set is empty.

The algorithm produces full well-formed dialogues, together with their annotations. The annotated dialogues can be used to train any standard dialogue state tracker.

3.5 Training Data Adaptations

We also synthesize new training data by adapting dialogues from domains with similar slots. For example, both restaurants and hotels have locations, so we can adapt a sentence like "find me a restaurant in the city center" to "find me a hotel in the city center". We substitute a matching domain noun phrase with the one for the new domain, and its slot values to those from the target ontology.

We also generate new multi-domain dialogues from existing ones. We use heuristics to identify the point where the domain switches and we concatenate single-domain portions to form a multi-domain dialogue.

4 Experimental Setting

4.1 The MultiWOZ Dataset

The MultiWOZ dataset (Budzianowski et al., 2018; Eric et al., 2019) is a multi-domain fully-labeled corpus of human-human written conversations. Its ontology has 35 slots in total from 7 domains. Each dialogue consists of a goal, multiple user and agent utterances, and annotations in terms of slot values at every turn. The dataset is created through crowdsourcing and has 3,406 single-domain and 7,032 multi-domain dialogues.

Of the 7 domains, only 5 have correct annotations and any data in the validation or test sets. Following Wu et al. (2019) we only focus on these 5 domains in this paper. The characteristics of the domains are shown in Table 2.

4.2 Machine Learning Models

We evaluate our data synthesis technique on two state-of-the-art models for the MultiWOZ dialogue state tracking task, TRADE (Wu et al., 2019) and SUMBT (Lee et al., 2019). Here we

	Attraction	Hotel	Restaurant	Taxi	Train
# user slots	3	10	7	4	6
# agent slots	5	4	4	2	2
# slot values	167	143	374	766	350
# real dialogues	3,469	4,196	4,836	1,919	3,903
# in-domain turns	10,549	18,330	18,801	5,962	16,081
# in-domain tokens	312,569	572,955	547,605	179,874	451,521
# domain subject templates	3	5	4	2	4
# slot name templates	15	17	21	18	16
# slot value templates	7	30	30	37	42
# information utterance templates	1	14	13	13	27
# synthesized dialogues	6,636	13,300	9,901	6,771	14,092
# synthesized turns	30,274	62,950	46,062	35,745	60,236
# synthesized tokens	548,822	1,311,789	965,219	864,204	1,405,201
transfer domain	Restaurant	Restaurant	Hotel	Train	Taxi
overlapping slots	2	6	6	4	4

Table 2: Characteristics of the MultiWOZ ontology, the MultiWOZ dataset, the template library, and the synthesized datasets for the zero-shot experiment on the 5 MultiWOZ domains. "user slots" refers to the slots the user can provide and the model must track, while "agent slots" refer to slots that the user requests from the agent (such as the phone number or the address). Note that total number of dialogues is smaller than the sum of dialogues in each domain due to multi-domain dialogues.

give a brief overview of each model; further details are provided in the respective papers.

TRADE TRAnsferable Dialogue statE generator (TRADE) uses a soft copy mechanism to either copy slot-values from utterance pairs or generate them using an Recurrent Neural Network (RNN) (Sutskever et al., 2014) decoder. This model can produce slot-values not encountered during training. The model is comprised of three main parts: an RNN utterance encoder which generates a context vector based on the previous turns of the dialogue; a slot-gate predictor indicating which (domain, slot) pairs need to be tracked, and a state generator that produces the final word distribution at each decoder time-step.

SUMBT Slot-Utterance Matching Belief Tracker (SUMBT) uses an attention mechanism over user-agent utterances at each turn to extract the slot-value information. It deploys a distance-based non-parametric classifier to generate the probability distribution of a slot-value and minimizes the log-likelihood of these values for all slot-types and dialogue turns. Specifically, their model includes four main parts: the BERT (Devlin et al., 2019) language model which encodes slot names, slot values, and utterance pairs, a multi-head attention module that computes an attention vector between slot and utterance representations, a RNN state tracking module, and a discriminative classifier which computes the probability of each slot value. The use of similarity to find relevant slot values makes the model depend on the ontology. Thus the model is unable to track unknown slot values.

4.3 Software and Hyperparameters

We used the Genie tool (Campagna et al., 2019) to synthesize our datasets. We incorporated our dialogue model and template library into a new version of the tool. The exact version of the tool used for the experiment, as well as the generated datasets, are available on GitHub[1]. For each experiment, we tuned the Genie hyperparameters separately on the validation set.

For the models, we use the code that was released by the respective authors, with their recommend hyperparameters. For consistency, we use the same data preprocessing to train both TRADE and SUMBT.

5 Experiments

5.1 Data synthesis

Our abstract transaction dialogue model has 13 abstract states, 15 agent dialogue acts, 17 user dialogue acts, and 34 transitions (Table 1). We have created 91 dialogue templates for this model. Dialogue templates were optimized using the validation data in the "Restaurant" domain.

[1] https://github.com/stanford-oval/zero-shot-multiwoz-acl2020

Model	Synth.	Joint	Slot Acc.
TRADE	no	44.2	96.5
	yes	43.0	96.4
SUMBT	no	46.7	96.7
	yes	46.9	96.6

Table 3: Accuracy on the full MultiWOZ 2.1 dataset (test set), with and without synthesized data.

We also created domain templates for each domain in MultiWOZ. The number of templates and other characteristics of our synthesis are shown in Table 2. To simulate a zero-shot environment in which training data is not available, we derived the templates from only the validation data of that domain. We did not look at in-domain training data to design the templates, nor did we look at any test data until the results reported here were obtained. In the table, we also include the domain we chose to perform domain adaptation (Section 3.5) and the number of slots from the adapted domain that are applicable to the new domain.

Note that the validation and test sets are the same datasets as the MultiWOZ 2.1 release.

5.2 Evaluation On All Domains

Our first experiment evaluates how our synthesized data affects the accuracy of TRADE and SUMBT on the full MultiWOZ 2.1 dataset. As in previous work (Wu et al., 2019), we evaluate the *Joint Accuracy* and the *Slot Accuracy*. Joint Accuracy measures the number of turns in which all slots are predicted correctly at once, whereas Slot Accuracy measures the accuracy of predicting each slot individually, then averages across slots. Slot Accuracy is significantly higher than Joint Accuracy because, at any turn, most slots do not appear, hence predicting an empty slot yields high accuracy for each slot. Previous results were reported on the MultiWOZ 2.0 dataset, so we reran all models on MultiWOZ 2.1.

Results are shown in Table 3. We observe that our synthesis technique, which is derived from the MultiWOZ dataset, adds no value to this set. We obtain almost identical slot accuracy, and our joint accuracy is within the usual margin of error compared to training with the original dataset. This is a sanity-check to make sure our augmentation method generates compatible data and training on it does not worsen the results.

5.3 Zero-Shot Transfer Learning

Before we evaluate zero-shot learning on new domains, we first measure the accuracy obtained for each domain when trained on the full dataset. For each domain, we consider only the subset of dialogues that include that particular domain and only consider the slots for that domain when calculating the accuracy. In other words, suppose we have a dialogue involving an attraction and a restaurant: a prediction that gets the attraction correct but not the restaurant will count as joint-accurate for the attraction domain. This is why the joint accuracy of individual domains is uniformly higher than the joint accuracy of all the domains. Table 4 shows that the joint accuracy for TRADE varies from domain to domain, from 50.5% for "Hotel" to 74.0% for "Train". The domain accuracy with the SUMBT model is better than that of TRADE by between 1% and 4% for all domains, except for "Taxi" where it drops by about 4.5%.

In our zero-shot learning experiment, we withhold all dialogues that refer to the domain of interest from the training set, and then evaluate the joint and slot accuracies in the same way as before. The joint accuracy with the TRADE model is poor throughout except for 59.2% for "Taxi". The rest of the domains have a joint accuracy ranging from 16.4% for "Restaurant" to 22.9% for "Train". Upon closer examination, we found that simply predicting "empty" for all slots would yield the same joint accuracy. The zero-shot results for SUMBT are almost identical to that of TRADE.

A different evaluation methodology is used by Wu et al. (2019) in their zero-shot experiment. The model for each domain is trained with the full dataset, except that all the slots involving the domain of interest are removed from the dialogue state. The slots for the new domain are present in the validation and test data, however. The method they use, which we reproduce here[2], has consistently higher slot accuracy, but slightly worse joint accuracy than our baseline, by 1.9% to 5.8%, except for "Taxi" which improves by 1% to 60.2%.

To evaluate our proposed technique, we add our synthesized data for the domain of interest to the training data in the zero-shot experiment. Besides synthesizing from templates, we also apply domain adaptation. The pairs of domain chosen for

[2]Wu et al. (2019) reported results on MultiWOZ 2.0, while we report MultiWOZ 2.1. The results on the two datasets are all within 3% of each other.

Model	Training	Attraction		Hotel		Restaurant		Taxi		Train	
		Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot
TRADE	Full dataset	67.3	87.6	50.5	91.4	61.8	92.7	72.7	88.9	74.0	94.0
	Zero-shot	22.8	50.0	19.5	62.6	16.4	51.5	59.2	72.0	22.9	48.0
	Zero-shot (Wu)	20.5	55.5	13.7	65.6	13.4	54.5	60.2	73.5	21.0	48.9
	Zero-shot (DM)	**34.9**	**62.2**	**28.3**	**74.5**	**35.9**	**75.6**	**65.0**	**79.9**	**37.4**	**74.5**
	Ratio of DM over full (%)	51.9	71.0	56.0	81.5	58.1	81.6	89.4	89.9	50.5	79.3
SUMBT	Full dataset	71.1	89.1	51.8	92.2	64.2	93.1	68.2	86.0	77.0	95.0
	Zero-shot	22.6	51.5	19.8	63.3	16.5	52.1	59.5	74.9	22.5	49.2
	Zero-shot (DM)	**52.8**	**78.9**	**36.3**	**83.7**	**45.3**	**82.8**	**62.6**	**79.4**	**46.7**	**84.2**
	Ratio of DM over full (%)	74.3	88.6	70.1	90.8	70.6	88.9	91.8	92.3	60.6	88.6

Table 4: Accuracy on the zero-shot MultiWOZ experiment (test set), with and without data augmentation. TRADE refers to Wu et al. (2019), SUMBT to Lee et al. (2019). "Zero-shot" results are trained by withholding in-domain data. "Zero-shot (Wu)" results are obtained with the unmodified TRADE zero-shot methodology, trained on MultiWOZ 2.1. "Zero-shot (DM)" refers to zero-shot learning using our Dialogue-Model based data synthesis. The last line of each model compares DM with full training, by calculating the % of the accuracy of the former to the latter.

adaptation are shown in Table 2, together with the number of slot names that are common to both domains. "Taxi" uses a subset of the slot names as "Train" but with different values. "Attraction", "Restaurant" and "Hotel" share the "name" and "area" slot; "Restaurant" and "Hotel" also share the "price range", "book day", "book time" and "book people" slots. For slots that are not shared, the model must learn both the slot names and slot values exclusively from synthesized data.

Our dialogue-model based zero-shot result, reported as "Zero-shot (DM)" in Table 4, shows that our synthesized data improves zero-shot accuracy on all domains. For TRADE, the joint accuracy improves between 6% on "Taxi" and 19% on "Restaurant", whereas for SUMBT, joint accuracy improves between 3% on "Taxi" and 30% on "Attraction". With synthesis, SUMBT outperforms TRADE by a large margin. Except for "Taxi" which has uncharacteristically high joint accuracy of 65%, SUMBT outperforms TRADE from 8% to 18%. This suggests SUMBT can make better use of synthesized data.

To compare synthesized with real training data, we calculate how close the accuracy obtained with the synthetic data gets to full training. We divide the accuracy of the former with that of the latter, as shown in the last row for each model in Table 4.

Overall, training with synthesized data is about half as good as full training for TRADE, but is 2/3 as good as for SUMBT (the ratio is 61% to 74%, ignoring "Taxi" as an outlier). This suggests that our synthesis algorithm is generating a reasonable variety in the dialogue flows; the pretrained BERT model, which imbues the model with general knowledge of the English language, is better

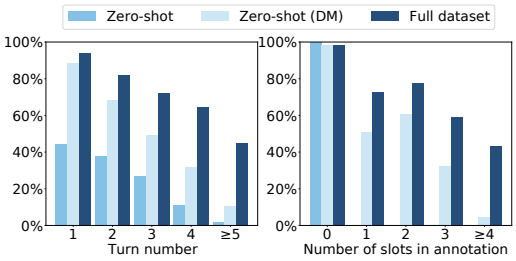

Figure 3: Breakdown of accuracy by turn number and number of slots of the TRADE model on the "Restaurant" domain. "Zero-shot" results are trained by withholding in-domain data, and "Zero-shot (DM)" is our data synthesis based on the Dialogue Model. "Full dataset" refers to training with all domains.

at compensating for the lack of language variety in synthesized data. Thus, the model only needs to learn the ontology and domain vocabulary from the synthesized data. Conversely, TRADE has no contextual pretraining and must learn the language from the limited dialogue data. This suggests that the combination of unsupervised pretraining and training on synthesized data can be effective to bootstrap new domains.

5.4 Error Analysis

To analyze the errors, we break down the result according to the turn number and number of slots in the dialogues in the test set, as shown in Fig. 3. We perform this analysis using the TRADE model on the "Restaurant" domain, which is the largest domain in MultiWOZ. We observe that the baseline model achieves 100% accuracy for turns with no slots, and 0% accuracy otherwise. The baseline results in the turn-number plot thus indicate

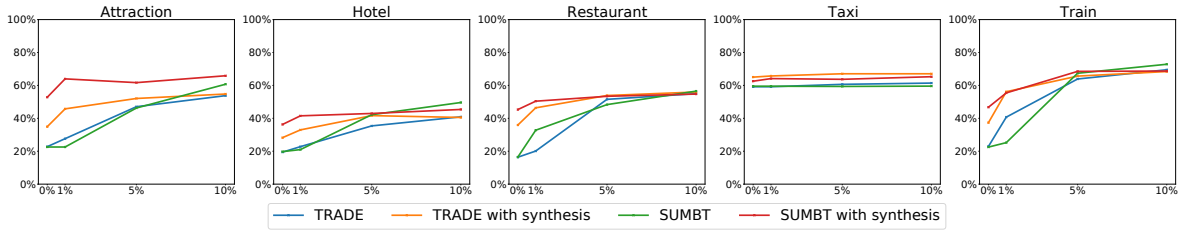

Figure 4: Accuracy plots for the few-shot MultiWOZ experiments. X axis indicates the percentage of real target domain data included in training. Y axis indicates joint accuracy.

the percentage of dialogues with all empty slots at each turn. It is possible for 5-turn dialogues to have all empty slots because a multi-domain dialogue may not have filled any slot in one domain.

By and large, the accuracy degrades for both the "full dataset" model and the "zero-shot (DM)" model, with the latter losing more accuracy than the former when there are 3 or 4 slots. The accuracy drops almost linearly with increasing turn numbers for the full model. This is expected because a turn is considered correct only if the full dialogue state is correct, and the state accumulates all slots mentioned up to that point. The results for the full and the zero-shot (DM) models look similar, but the zero-shot model has a larger drop in later turns. Modeling the first few turns in the dialogue is easier, as the user is exclusively providing information, whereas in later turns more interactions are possible, some of which are not captured well by our dialogue model.

5.5 Few-Shot Transfer Learning

Following Wu et al. (2019), we also evaluate the effect of mixing a small percentage of real training data in our augmented training sets. We use a naive few-shot training strategy, where we directly add a portion of the original training data in the domain of interest to the training set.

Fig. 4 plots the joint accuracy achieved on the new domain with the addition of different percentages of real training data. The results for 0% are the same as the zero-shot experiment. The advantage of the synthesized training data decreases as the percent of real data increases, because real data is more varied, informative, and more representative of the distribution in the test set. The impact of synthesized data is more pronounced for SUMBT than TRADE for all domains even with 5% real data, and it is significant for the "Attraction" domain with 10% real data. This suggests that SUMBT needs more data to train, due to having more parameters, but can utilize additional synthesized data better to improve its training.

6 Conclusion

We propose a method to synthesize dialogues for a new domain using an abstract dialogue model, combined with a small number of domain templates derived from observing a small dataset. For transaction dialogues, our technique can bootstrap new domains with less than 100 templates per domain, which can be built in a few person-hours. With this little effort, it is already possible to achieve about 2/3 of the accuracy obtained with a large-scale human annotated dataset. Furthermore, this method is general and can be extended to dialogue state tracking beyond transactions, by building new dialogue models.

We show improvements in joint accuracy in zero-shot and few-shot transfer learning for both the TRADE and BERT-based SUMBT models. Our technique using the SUMBT model improves the zero-shot state of the art by 21% on average across the different domains. This suggests that pretraining complements the use of synthesized data to learn the domain, and can be a general technique to bootstrap new dialogue systems.

We have released our algorithm and dialogue model as part of the open-source Genie toolkit, which is available on GitHub[3].

Acknowledgments

This work is supported in part by the National Science Foundation under Grant No. 1900638; Mehrad Moradshahi is supported by a Stanford Graduate Fellowship.

[3] https://github.com/stanford-oval/genie-toolkit

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Harry Bunt, Volha Petukhova, and Alex Chengyu Fang. 2017. Revisiting the ISO standard for dialogue act annotation. In *Proceedings of the 13th Joint ISO-ACL Workshop on Interoperable Semantic Annotation (ISA-13)*.

Giovanni Campagna, Silei Xu, Mehrad Moradshahi, Richard Socher, and Monica S. Lam. 2019. Genie: A generator of natural language semantic parsers for virtual assistant commands. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, pages 394–410, New York, NY, USA. ACM.

Jianpeng Cheng, Siva Reddy, and Mirella Lapata. 2018. Building a neural semantic parser from a domain ontology. *arXiv preprint arXiv:1812.10037*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. MultiWOZ 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 360–365. IEEE.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017a. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017b. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual. Technical report, University of Colorado, Institute of Cognitive Science.

John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2879–2888.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. Sumbt: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking model. *arXiv preprint arXiv:1812.00899*.

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

Yu Su, Ahmed Hassan Awadallah, Madian Khabsa, Patrick Pantel, Michael Gamon, and Mark Encarnacion. 2017. Building natural language interfaces to web APIs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17*. ACM Press.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342. Association for Computational Linguistics.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

Jason D Williams, Matthew Henderson, Antoine Raux, Blaise Thomson, Alan Black, and Deepak Ramachandran. 2014. The dialog state tracking challenge series. *AI Magazine*, 35(4):121–124.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1457.

Silei Xu, Giovanni Campagna, Jian Li, and Monica S Lam. 2020. Schema2qa: Answering complex queries on the structured web with a neural model. *arXiv preprint arXiv:2001.05609*.

Dian Yu and Zhou Yu. 2019. MIDAS: A dialog act annotation scheme for open domain human machine spoken conversations. *arXiv preprint arXiv:1908.10023*.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, et al. 2019. CoSQL: A conversational Text-to-SQL challenge towards cross-domain natural language interfaces to databases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1962–1979.

Tiancheng Zhao and Maxine Eskenazi. 2018. Zero-shot dialog generation with cross-domain latent actions. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 1–10.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. 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)*, pages 1458–1467.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*.