Zero and Few-Shot Localization of Task-Oriented Dialogue Agents with a Distilled Representation

Mehrad Moradshahi **Sina J. Semnani** **Monica S. Lam**
Computer Science Department
Stanford University
Stanford, CA
{mehrad,sinaj,lam}@cs.stanford.edu

Abstract

Task-oriented Dialogue (ToD) agents are mostly limited to a few widely-spoken languages, mainly due to the high cost of acquiring training data for each language. Existing low-cost approaches that rely on cross-lingual embeddings or naive machine translation sacrifice a lot of accuracy for data efficiency, and largely fail in creating a usable dialogue agent. We propose automatic methods that use ToD training data in a source language to build a high-quality functioning dialogue agent in another target language that has no training data (i.e. zero-shot) or a small training set (i.e. few-shot). Unlike most prior work in cross-lingual ToD that only focuses on Dialogue State Tracking (DST), we build an end-to-end agent.

We show that our approach closes the accuracy gap between few-shot and existing full-shot methods for ToD agents. We achieve this by (1) improving the dialogue data representation, (2) improving entity-aware machine translation, and (3) automatic filtering of noisy translations.

We evaluate our approach on the recent bilingual dialogue dataset BiToD. In Chinese to English transfer, in the zero-shot setting, our method achieves 46.7% and 22.0% in Task Success Rate (TSR) and Dialogue Success Rate (DSR) respectively. In the few-shot setting where 10% of the data in the target language is used, we improve the state-of-the-art by 15.2% and 14.0%, coming within 5% of full-shot training.[1]

1 Introduction

While dialogue agents in various forms have become commonplace in parts of the world, their lack of support for most human languages has prevented access to the benefits they provide for much of the world. Commercial virtual assistants for example, only support a handful of languages, as extending their functionality to each new language is extremely costly, partially due to the need for collecting new annotated training data in that language.

In recent years, several non-English task-oriented dialogue (ToD) datasets have been created; they are either collected from scratch such as RiSAWOZ (Quan et al., 2020) and CrossWOZ (Zhu et al., 2020), paraphrased from synthetic sentences by crowdworkers such as BiToD (Lin et al., 2021), or manually translated from another language (Li et al., 2021b). All of these approaches are labor-intensive, expensive, and time-consuming; such investment is unlikely to be made for less widely spoken languages.

Cross-lingual transfer, i.e. using training data from other languages to build a dialogue agent for a specific language, seems especially appealing. An emerging line of work has employed machine translation of training data, and multilingual pre-trained neural networks to tackle this task (Sherborne et al., 2020; Li et al., 2021a; Moradshahi et al., 2023). However, work in ToD cross-lingual transfer has for the most part, focused on understanding the user input, namely Dialogue State Tracking (DST) and Natural Language Understanding (NLU). Other necessary parts of a dialogue agent like policy and response generation have mostly remained unexplored.

In this paper, we present a methodology for building a fully functional dialogue agent for a new language (e.g. English), by using training data in another language (e.g. Chinese) with little to no additional manual dataset creation effort. We found that despite prior efforts to improve modeling for existing ToD datasets, the dialogue representation used as input to these models, e.g. full dialogue history in natural language (Hosseini-Asl et al., 2020), is sub-optimal, especially when the training data is either scarce or created automatically us-

[1] Code can be accessed at https://github.com/stanford-oval/dialogues

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 886–901
May 2-6, 2023 ©2023 Association for Computational Linguistics

ing noisy machine translation. We propose a new *Distilled* representation to fix the shortcomings of current representations. We also found that previously proposed entity-aware translation technique Moradshahi et al. (2023) to be inadequate. Our proposed technique effectively combines entity-aware neural machine translation with text similarity classifiers to automatically create training data for a new language. This paper explains all the ingredients we found useful, and motivates their use by performing extensive ablation studies.

The contributions of this paper are:

1. *A new state-of-the-art result for the BiToD dataset in both few-shot and full-shot settings on English* according to all of our 6 automatic metrics, including an improvement of 14.0% and 2.9%, respectively, in Dialogue Success Rate (DSR). In fact, using our Distilled representation, our few-shot model trained on only 10% of the training data, achieves similar results to the previous SOTA model trained on 100% training data.
2. *The first dialogue agent created in the zero-shot cross-lingual transfer setting*, i.e. starting from no training data in the target language. Our agent achieves 71%, 62%, 40%, and 47% of the performance of a full-shot agent in terms of Joint Goal Accuracy (JGA), Task Success Rate (TSR), DSR, and BLEU score, respectively.
3. *A concise dialogue representation designed for cross-lingual ToD agents*. The Distilled dialogue representation works well with our new decomposition of agent subtasks, making significant improvements possible.
4. *An improved methodology for automatic translation of ToD training data*. We adapt and improve an existing entity-aware machine translation system that localizes entities (Moradshahi et al., 2023), extend it to agent response generation, and equip it with a filtering step that increases the quality of the resulting translations.

2 Related Work

2.1 Multilingual Dialogue Datasets

MultiWOZ (Budzianowski et al., 2018; Ramadan et al., 2018; Eric et al., 2019) and CrossWOZ (Zhu et al., 2020) are two monolingual Wizard-Of-Oz dialogue datasets that cover several domains, suitable for building travel dialogue agents in English and Chinese respectively. For the 9th Dialog System Technology Challenge (DSTC-9) (Gunasekara et al., 2020), they were translated to Chinese and English using Google Translate.

GlobalWOZ (Ding et al., 2021), AllWOZ (Zuo et al., 2021), and Multi2WOZ (Hung et al., 2022) translate MultiWOZ to even more languages such as Spanish, Hindi, and Indonesian, with human translators post-editing machine translated dialogue templates, and filling them with newly collected local entities. Although manual post-editing improves data quality and ensures fluency, it also increases the cost and time to create new datasets, thus limiting scalability.

Different from these translation approaches, Lin et al. (2021) introduced BiToD, the first bilingual dataset for *end-to-end* ToD modeling. BiToD uses a dialogue simulator to generate dialogues in 5 tourism domains in English and Chinese, then uses crowdsourcing to paraphrase entire dialogues to be more natural. Unlike WOZ-style datasets which usually suffer from poor annotation quality due to human errors (Moradshahi et al., 2023), BiToD is automatically annotated during synthesis. Since neither manual nor machine translation is used in the creation of BiToD, it does not contain translationese (Eetemadi and Toutanova, 2014) or other artifacts of translated text (Clark et al., 2020), and provides a realistic testbed for cross-lingual transfer of task-oriented dialogue agents.

2.2 Multilingual Dialogue State Tracking

Mrkšić et al. (2017) proposed using cross-lingual word embeddings for zero-shot cross-lingual transfer of DST models. With the advent of large language models, contextual embeddings obtained from pre-trained multilingual language models (Devlin et al., 2018; Xue et al., 2021; Liu et al., 2020) have been used to enable cross-lingual transfer in many natural language tasks, including DST.

Chen et al. (2018) used knowledge distillation (Hinton et al., 2015) to transfer DST capabilities from a teacher DST model in the source language to a student model in the target language.

Machine translation has been used for DST, both as a way of obtaining cross-lingual representations, and to translate training data. For instance, Schuster et al. (2019) used representations obtained from machine translation models and reported that it performs better than training with machine translated training data for single-turn commands. More advanced data translation approaches like the entity-aware method of Moradshahi et al. (2023) further

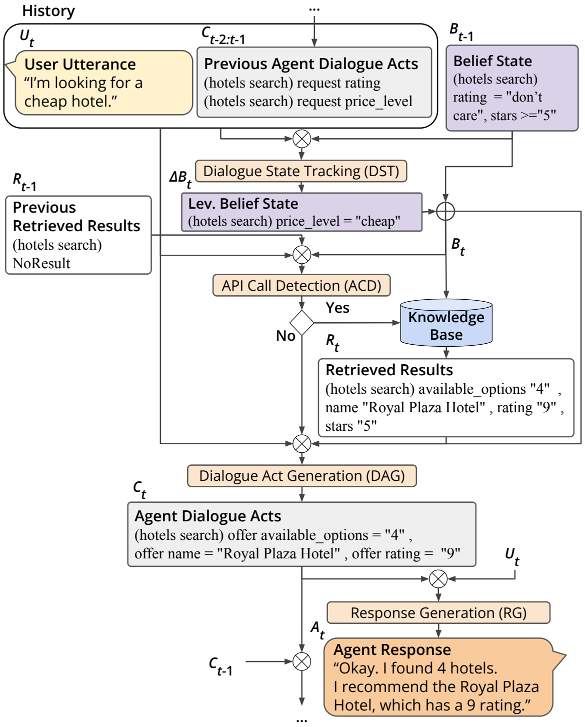
History
...
U_t
User Utterance
"I'm looking for a cheap hotel."
$C_{t-2:t-1}$
Previous Agent Dialogue Acts
(hotels search) request rating
(hotels search) request price_level
B_{t-1}
Belief State
(hotels search)
rating = "don't care", stars >="5"
Dialogue State Tracking (DST)
ΔB_t
Lev. Belief State
(hotels search) price_level = "cheap"
R_{t-1}
Previous Retrieved Results
(hotels search)
NoResult
B_t
API Call Detection (ACD)
Yes
No
Knowledge Base
R_t
Retrieved Results
(hotels search) available_options "4" , name "Royal Plaza Hotel" , rating "9" , stars "5"
Dialogue Act Generation (DAG)
C_t
Agent Dialogue Acts
(hotels search) offer available_options = "4" ,
offer name = "Royal Plaza Hotel" , offer rating = "9"
U_t
Response Generation (RG)
A_t
Agent Response
"Okay. I found 4 hotels.
I recommend the Royal Plaza Hotel, which has a 9 rating."
C_{t-1}
...

Figure 1: Inference-time flow diagram for our dialogue agent. DST, ACD, DAG, and RG share the same neural model. U, A, C, B, and R indicate user utterance, agent response, agent dialogue acts, dialogue state, and retrieved database results respectively. t is the turn number. $\otimes$ indicates text concatenation. $\oplus$ refers to the update rule in Equation 1.

improved the DST data quality achievable with machine translation.

3 Distilled ToD Agent

Our methodology includes a dialogue task decomposition and a Distilled dialogue representation that are tailored to cross-lingual ToD agents. In this section we describe these two components.

We follow the end-to-end task-oriented dialogue (ToD) setting (Hosseini-Asl et al., 2020) where a user converses freely with an agent over several turns to accomplish his/her goal with all of its constraints (e.g. "book a restaurant that is rated at least 3."). In each turn, the agent must access its database if needed to find the requested information (e.g. find a restaurant that satisfies user constraints), decide on an action (e.g. to present the information to the user or to ask follow-up questions) and finally respond to the user in natural language based on the action it selects.

3.1 Preliminaries

Formally, a *dialogue* $D = \{U_1, A_1, ..., U_T, A_T\}$ is a set of alternating user utterances U_t and agent responses A_t for a number of turns T.

A *belief state* at turn t, B_t, consists of a list of $\langle domain, intent \rangle$ tuples and a set of $\langle slot, relation, value \rangle$ tuples. *Intent* is the user intent, either search or book. *Relation* is a comparison or membership operator. *Value* can be one or more entity names or strings from the ontology, or a literal. To see all possible domains, slots and values please refer to Table 4 in Lin et al. (2021).

The *Levenshtein belief state* (Lin et al., 2020) is the difference between belief states in consecutive turns, i.e. $\Delta B_t = B_t - B_{t-1}$. It captures only the relations and values that have changed in the last user utterance, or tuples that have been added or removed.

An *Agent dialogue act* at turn t, C_t, is a list of $\langle domain, intent \rangle$ tuples and a set of $\langle dialogue\_act\_name, slot, value \rangle$ tuples indicating the action the agent takes and the information offered to the user, if any.

3.2 Task Decomposition

The task of dialogue agents is usually broken down to subtasks, which may be performed by a pipelined system (Gao et al., 2018) or by a single neural network (Hosseini-Asl et al., 2020; Lei et al., 2018). Here we describe our subtasks and their inputs and outputs (Figure 1).

After the user speaks at turn t, the agent has access to the belief state up to the previous turn (B_{t-1}), the history of agent dialogue acts ($C_1, ..., C_{t-1}$), and the history of agent and user utterances so far ($A_1, ..., A_{t-1}$ and $U_1, ..., U_t$). Our agent performs the following four subtasks:

1. *Dialogue State Tracking (DST)*: Generate ΔB_t, the Levenshtein belief state, for the current turn based on the previous belief state, the last two agent dialogue acts[2], and the current user utterance. ΔB_t is combined with B_{t-1} to produce the current belief state.

$$\begin{aligned} \Delta B_t &= \mathrm{DST}(B_{t-1}, C_{t-2}, C_{t-1}, U_t) \\ B_t &\leftarrow B_{t-1} + \Delta B_t \end{aligned} \tag{1}$$

2. *API Call Detection (ACD)*: Call an API to query the database, if needed.

$$q_t = \mathrm{ACD}(B_t, C_{t-2}, C_{t-1}, U_t, R_{t-1}) \tag{2}$$

$$R_t \leftarrow q_t ?\, \mathrm{KB}(B_t) \ : \ \emptyset \tag{3}$$

[2] Our ablation study described in Section 6.1 justifies the use of the last two agent dialogue acts instead of just the last one.

In turn t, ACD determines if an API call is necessary. If so, the result R_t is the top entity in the knowledge base KB, based on a deterministic ranking scheme, that matches the API call constraints in B_t, and is empty otherwise. If no entities match the constraint, we set R_t to the special value NORESULT.

3. *Dialogue Act Generation (DAG)*: Generate C_t, the agent dialogue act for the current turn based on the current belief state, the last two agent dialogue acts, the user utterance, and the result from the API call.

$$C_t = \mathrm{DAG}(B_t, C_{t-2}, C_{t-1}, U_t, R_t) \quad (4)$$

4. *Response Generation (RG)*: Convert the agent dialogue act C_t to the new agent utterance A_t. Note that C_t contains all the necessary information for this subtask. However, providing U_t improves response fluency and choice of words, leading to a higher BLEU score, partly due to mirroring (Kale and Rastogi, 2020).

$$A_t = \mathrm{RG}(U_t, C_t) \quad (5)$$

3.3 The Distilled Dialogue Representation

The design of Distilled is based on the following principles:

1. For cross-lingual agents, it is important to reduce the impact of translation errors. The representation should make minimal use of natural language by using a formal representation where possible.
2. Dialogues can get long, but the representation should be succinct, containing only the necessary information, so the neural network need not *learn* to ignore unnecessary information from copious data. This improves data efficiency as well as the training and inference speed of neural models.

We note that BiToD's original representation (Lin et al., 2021) follows neither of these principles.[3] It makes extended use of natural language: all previous user and agent natural language utterances are included in the input of all subtasks. It has many redundancies: for each subtask, it inputs the concatenation of all previous subtask's inputs and outputs. In the following, we highlight the changes we made to the (Lin et al., 2021) representation.

Replace agent utterances with formal agent dialogue acts. Since agent responses are automatically generated, it is possible to capture all information useful to the different subtasks with formal agent dialogue acts. In this way, the neural network need not interpret previous natural language utterances.

We take two steps to generate the agent responses: DAG (Dialogue Act Generation) first produces the formal act, C_t, which is then fed into RG (Response Generation) to generate the natural language response A_t. Note that RG is not a part of the dialogue loop: the natural language A_t only serves to communicate to the user; it is the formal C_t from DAG that gets fed to subsequent subtasks instead. In contrast, Lin et al. (2021) generates the agent response directly from API results. Hosseini-Asl et al. (2020) also separates the response generation into two steps, but they use A_t instead of C_t as input to the semantic parser for the next turn.

Note that the agent dialogue acts are independent of the natural language used in the dialogues, if we ignore the entity values. This is beneficial to cross-lingual agents as it can learn easier from data available in other languages. Furthermore, DAG can be validated on whether the output dialogue acts match the gold answers exactly. This is not possible with natural language results, whose quality is typically estimated with BLEU score.

Shorten user utterance history. Since the belief state formally summarizes what the user has said, we remove previous user utterances $U_1, ..., U_{t-1}$ from input to all subtasks, relying on the belief state B_{t-1} instead.

Untangle API call detection from response generation. After DST is done, depending on whether or not an API call is needed, Lin et al. (2021) either directly generates the agent response, or makes the API call and then generates the response in two steps. Our design is to always take two steps: (1) generate the API call *or indicate that there is none*, and (2) generate the agent response.

4 Automatic Dialogue Data Translation

Given a training dataset for one language, we automatically generate a training set in the target language we are interested in. This problem has been studied in the context of NLU for questions (Moradshahi et al., 2020; Sherborne et al., 2020; Li et al., 2021a) and for dialogues (Moradshahi et al., 2023;

[3] We found this to be true for several previously-proposed popular representations of MultiWOZ as well (Lei et al., 2018; Chen et al., 2019).

Ding et al., 2021; Zuo et al., 2021). One challenge is that the translated dataset should refer to entities in the target language. Thus, Moradshahi et al. (2020) proposed to first use cross-attention weights of the neural translation model to align entities in the original and translated sentences, then replace entities in the translated sentences with local entities from a target language knowledge base. Our initial experiments showed that applying this approach directly to end-to-end dialogue datasets does not yield good performance, especially for response generation. Thus, we adapted and improved this approach for dialogues as discussed below.

4.1 Alignment for Dialogues

First, we found that while translation with alignment works for NLU, it does not work well for RG. Machine translation introduces two kinds of error: (1) Translated sentences can be ungrammatical, incorrect, or introduce spurious information. (2) The alignment for entities may be erroneous, which can seriously hurt the factual correctness of the responses. As shown in Moradshahi et al. (2023), these errors are tolerable in NLU since (1) sentences are seen by machines, not shown to users, (2) pre-trained models like mBART are somewhat robust to noisy inputs, since they are pre-trained on perturbed data. However, training with such low-quality data is not acceptable for RG, since the learned responses are shown directly to the user.

Second, we found alignment recall to be particularly low for an important category: entities that are mostly quantitative. We observe that dates, times, and prices can be easily mapped between different languages using rules. We propose to first try to translate such entities with dictionaries such as those available in dateparser (Scrapinghub, 2015) and num2words (faire Linux, 2017), and to match them in the translated text. We resort to using neural alignment only if no such match is found.

4.2 Filtering Translation Noise for RG

To reduce translation noise for RG, we automatically filter the translated data based on the semantic textual similarity between the source and translated sentences. For this purpose, we use LaBSE (Feng et al., 2020), a multilingual neural sentence encoder based on multilingual BERT (Devlin et al., 2018), trained on translation pairs in various languages with a loss function that encourages encoding pairs to similar vectors. To score a pair of sentences, the model first calculates an embedding for each sentence and computes the cosine distance between those vectors. The lower the distance is, the more semantically similar the sentences are, according to the model.

In creating the RG training set, we first translate the source agent utterances to the target language and use LaBSE to remove pairs whose similarity score is below a threshold. We found a threshold of 0.8 to work best empirically. Higher thresholds would inadvertently filter correctly translated utterances. We construct the final training data by pairing aligned translated utterances that pass the filter with their corresponding translated agent dialogue acts.

5 Experiment Setting

5.1 Base Dataset

We perform our experiments on BiToD, a large-scale high-quality bilingual dataset created using the Machine-to-Machine (M2M) approach. It is a multi-domain dataset, including restaurants, hotels, attractions, metro, and weather domains. It has a total of 7,232 dialogues (3,689 dialogues in English and 3,543 dialogues in Chinese) with 144,798 utterances in total. The data is split into 5,787 dialogues for training, 542 for validation, and 902 for testing. The training data is from the same distribution as validation and test data.

5.2 Implementation details

Our code is implemented in PyTorch (Paszke et al., 2019) using GenieNLP (Campagna et al., 2019) library for training and evaluation metrics. We also use the Dialogues[4] library for data preprocessing and evaluation. We use pre-trained models available through HuggingFace's Transformers library (Wolf et al., 2019). The following model names are from that library. We use *mbart-large-50* as the neural model for our agent in all our experiments. All models use a standard Seq2Seq architecture with a bidirectial encoder and left-to-right autoregressive decoder. mBART is pre-trained to denoise text in 50 languages, while mT5 is trained on 101 languages. mBART uses sentence-piece (Kudo and Richardson, 2018) for tokenization.

In each setting, all four subtasks of DST, API detection, dialogue act generation, and response generation are done in a single model, where we specify the task by prepending a special token to the

[4] https://github.com/stanford-oval/dialogues

input. We found mBART to be especially effective in zero-shot settings as the language of its outputs can be controlled by providing a language-specific token at the beginning of decoding. Additionally, its denoising pre-training objective improves its robustness to the remaining translation noise.

For translation, we use the publicly available *mbart-large-50-many-to-one-mmt* (~611M parameters) model which can directly translate text from any of the 50 supported languages to English. It is an mBART model additionally fine-tuned to do translation. We use greedy decoding and train our models using teacher-forcing and token-level cross-entropy loss. We used Adam (Kingma and Ba, 2014) as our optimizer with a starting learning rate of 2×10^{-5} and linear scheduling. These hyperparameters were chosen based on a limited hyperparameter search on the validation set. For the numbers reported in the paper, due to cost, we performed only a single run for each experiment.

Our models were trained on virtual machines with a single NVIDIA V100 (16GB memory) GPU on the AWS platform. For a fair comparison, all monolingual models were trained for the same number of iterations of 60K, and bilingual models for 120K. In the few-shot setting, we fine-tuned the model for 3K steps on 1% of the data and 6K steps on 10% of the data. Sentences are batched based on their input and approximate output token count for better GPU utilization. We set the total number of tokens per batch to 800 for mBART. Due to the verbosity and redundancy of the original BiToD representation, Lin et al. (2021) used a batch size of 1 example for training mbart-large. Using our Distilled representation, however, we can fit up to 6 examples in each batch and process each batch 3 times faster during training. Training and evaluating each model takes about 10 GPU-hours on average.

During error analysis, we noticed that although certain slots (max_temp and min_temp slots in Metro domain, and time and price_range slots in Weather domain) are present in the retrieved knowledge base values, the model does not learn to output them in the agent dialogue act generation subtask. This issue stems from BiToD's non-deterministic policy where an agent sometimes provides these slots and sometimes not in the gold training data. To mitigate this, during evaluation, we automatically check if these slots are present in the input and append them and their retrieved values to the generated agent dialogue acts.

At inference time, we use the predicted belief state as input to subsequent turns instead of ground truth. However, to avoid the conversation from diverging from its original direction, Lin et al. (2021) use the ground-truth natural-language agent response as input for the next turn. To make sure the settings are equivalent for a fair comparison, we use ground-truth agent acts as input for the next turn.

5.3 Evaluation Metrics

We use the following metrics to compare different models. Scores are averaged over all turns unless specified otherwise.

- **Joint Goal Accuracy (JGA)** (Budzianowski et al., 2018): Is the standard metric for evaluating DST. JGA for a dialogue turn is 1 if all slot-relation-value triplets in the generated belief state match the gold annotation, and is 0 otherwise.

- **Task Success Rate (TSR)** (Lin et al., 2021): A task, defined as a pair of domain and intent, is completed successfully if the agent correctly provides all the user-requested information and satisfies the user's initial goal for that task. TSR is reported as an average over all tasks.

- **Dialogue Success Rate (DSR)** (Lin et al., 2021): DSR is 1 for a dialogue if all user requests are completed successfully, and 0 otherwise. DSR is reported as an average over all dialogues. We use this as the main metric to compare models, since the agent needs to complete all dialogue subtasks correctly to obtain a full score on DSR.

- **API** (Lin et al., 2021): For a dialogue turn, is 1 if the model correctly predicts to make an API call, and all the constraints provided for the call match the gold. It is 0 otherwise.

- **BLEU** (Papineni et al., 2002): Measures the natural language response fluency based on n-gram matching with the human-written gold response. BLUE is calculated at the corpus level.

- **Slot Error Rate (SER)** (Wen et al., 2015): It complements BLEU as it measures the factual correctness of natural language responses. For each turn, it is 1 if the response contains all entities present in the gold response, and is 0 otherwise.

Representation	JGA ↑	TSR ↑	DSR ↑	API ↑	BLEU ↑	SER ↓
Original (Lin et al., 2021)	69.19	69.13	47.51	67.92	38.48	14.93
Distilled (Ours)	**76.79**	**75.64**	**53.39**	**76.33**	**42.54**	**10.61**
• Generate full state	74.30	74.19	50.90	73.93	41.90	11.38
• Natural agent response	75.62	73.41	49.10	73.93	40.94	11.90
• Only last agent turn	73.97	74.19	52.71	74.27	41.83	11.81
• Prev. user utterance as state	71.75	61.66	33.94	67.67	39.72	15.97
• Remove state	70.84	51.89	24.43	66.47	37.10	19.61

Table 1: Full-shot English monolingual training with ablation. All results are reported on the English test set of BiToD using the same evaluation script. The best result is in bold.

6 Results and Discussion

We first show how our Distilled representation affects the performance of an agent in a full-shot setting. We then evaluate our proposed techniques on cross-lingual settings with varying amounts of available training data.

6.1 Evaluation of Distilled Representation

To understand how our design of Distilled representation affects the performance of ToD agents in general, we train an English agent using all the English training data and perform an ablation study (Table 1). We observe that even though the Distilled representation removes a lot of natural language inputs, it improves the best previous English-only results on JGA, TSR, DSR, API, BLEU and SER by 7.6%, 6.5%, 5.9%, 8.4%, 4.1%, and 4.7%, respectively. This suggests that natural language utterances carry a lot of redundant information, and the verbosity may even hurt the performance. Note that the improvement in BLEU is also accompanied by an improvement of factuality measured by SER.

Furthermore, using the Distilled representation reduces training time by a factor of 3. See Section 5.2 for more details.

Generate full state. Our first ablation study confirms that the proposal by Lin et al. (2020) to predict the Levenshtein belief state (ΔB_t) is indeed better than the cumulative state (B_t). Note that the training time per gradient step is more than twice as long in this ablation since the outputs are longer.

Natural agent response. Here we use natural language agent responses as input instead of agent dialogue acts, replacing C_{t-1}, C_{t-2} with A_{t-1}, A_{t-2}. The drop in TSR and DSR shows this is an important design choice - distilling natural language into a concise formal representation improves the model's ability to understand the important information in the sentence.

Only last agent turn. When we remove C_{t-2} from the input and only use C_{t-1}, we observe a drop across all metrics. This is because some turns in BiToD refer to the agent's states from two turns ago. We experimented with carrying three turns, but there was no improvement.

Previous user utterance as state. In this ablation, we use U_{t-1} instead of B_{t-1} as subtask inputs. Compared to all previous ablations, accuracy drastically decreases across all metrics, especially JGA. This is expected since the information from earlier turns present in the dialogue state is now lost. Additionally, it shows that the dataset is highly contextual and therefore a summary of the conversation history is necessary.

Remove state. We remove B_{t-1} without adding back the previous user utterance U_{t-1}. Compared to the previous ablation, TSR and DSR drop by 10.5% and 5.2% respectively. This difference shows U_{t-1} does contain part of the information captured in B_{t-1}.

6.2 Evaluation of Cross-Lingual Transfer

The goal of this experiment is to create an agent in a *target* language, given the full training data in a source language ($\mathcal{D}_{\text{src}}$), and a varying amount of training data in a target language ($\mathcal{D}_{\text{tgt}}$). We also assume that valuation and test data are available in both source and target languages. We chose Chinese as the source language and English as the target language so we can perform error analysis and the model outputs are understandable for a wider audience.

6.2.1 Varying Target Training Data

Full-Shot. In the full-shot experiments, all of $\mathcal{D}_{\text{tgt}}$ is available for training. We train two models on two data sets: (1) on a shuffled mix of $\mathcal{D}_{\text{src}}$ and D_{tgt}. (2) on $\mathcal{D}_{\text{tgt}}$ alone. The ablation "*−Mixed*" in Table 2 refers to the latter.

Zero-Shot. In our zero-shot experiments, we train with a canonicalized $\mathcal{D}$SRC and an automatically translated data set, as explained below.

Setting	JGA ↑	TSR ↑	DSR ↑	API ↑	BLEU ↑	SER ↓
			Full-Shot			
MinTL(mT5)	72.16	71.18	51.13	71.87	40.71	13.75
− Mixed	69.19	69.13	47.51	67.92	38.48	14.93
MinTL(mBART)	69.37	42.45	17.87	65.35	28.76	–
− Mixed	67.36	56.00	33.71	57.03	35.34	–
Ours	**77.52**	75.04	**54.07**	74.44	41.46	11.17
− Mixed	76.79	**75.64**	53.39	**76.33**	**42.54**	**10.61**
			Zero-Shot			
Ours	**55.33**	**46.74**	**21.95**	**63.04**	**20.01**	**20.52**
− Filtering	54.83	45.03	19.68	60.81	19.11	20.86
− Alignment	47.21	4.72	1.13	52.74	8.26	39.20
− Translation	14.73	3.52	1.58	6.26	0.69	41.30
− Canonicalization	2.13	1.20	0.00	0.26	0.25	42.39
			Few-Shot (1%)			
Ours	**64.60**	**57.89**	**34.16**	**62.09**	**28.15**	**17.94**
− Filtering	63.88	57.80	32.35	59.95	28.00	18.57
− Alignment	58.86	51.89	23.76	57.12	26.84	21.56
− Translation	49.58	41.34	19.68	46.05	22.73	24.86
− Canonicalization	44.56	42.97	20.36	46.23	23.08	24.77
Few-shot Only	25.08	24.61	11.09	23.67	18.71	32.62
			Few-Shot (10%)			
MinTL(mT5)	58.85	56.43	34.16	57.54	31.20	–
− Translation	48.77	44.94	24.66	47.60	29.53	19.75
Few-shot Only	19.86	6.78	1.36	17.75	10.35	–
MinTL(mBART)	37.50	21.61	10.18	27.44	17.86	–
− Translation	42.84	36.19	16.06	41.51	22.50	–
Few-shot Only	4.64	1.11	0.23	0.60	3.17	–
Ours	**72.70**	**71.61**	**48.19**	**72.56**	**36.02**	**12.71**
− Filtering	72.45	69.55	44.57	69.55	34.67	13.62
− Alignment	68.40	63.38	38.24	63.38	32.99	16.63
− Translation	67.13	63.12	41.40	63.64	32.86	16.40
− Canonicalization	64.51	63.64	40.27	62.69	32.71	16.63
Few-shot Only	57.18	54.80	28.73	55.66	29.61	19.66

Table 2: All results are reported on the original English test set of BiToD using the same evaluation script. The best result in each section is in bold. Each "−" removes one additional component from the previous row. All MinTL results are from Lin et al. (2021). SER numbers are not available for some models. An upward arrow is show for columns where bigger numbers are better, and vice versa.

Canonicalization: To increase transfer learning from the source to the target language, we use the same canonical formal representation across languages (Moradshahi et al., 2020; Razumovskaia et al., 2021). To do so, we adapt $\mathcal{D}$src so that the domain names, slot names, agent dialogue acts, and API names in the formal representation to be the same as the target language. Note that the user utterance, agent response, and slot values will remain in the source language. The BiToD dataset has a one-to-one mapping for most of the above and we added the missing items.

Translation: We use machine translation to convert the user and agent utterances and slot values in $\mathcal{D}_{\text{src}}$ to create a training set for the target language.

Alignment: After translating the data, we use alignment (Section 4) to localize entities while ensuring the entities in translated utterances still match the values specified in annotations.

Filtering: We use the filtering procedure described in Section 4.2 to remove turns where agent responses are deemed to have low translation quality.

In Table 2, *Ours* refers to our main approach, which combines all four techniques. Each ablation incrementally takes away one of the techniques.

Few-Shot. In the few-shot setting, we start with our pre-trained zero-shot models (with various ablations) and further fine-tune it on 1% and 10% of $\mathcal{D}_{\text{tgt}}$, which comprises 29 and 284 dialogues, respectively. Lin et al. (2021) reported the results only for the 10% setting. We use their few-shot data split in that case to be directly comparable. We add one more ablation study where we eliminate cross-lingual transfer by training a model only on the few-shot data (Few-shot Only).

6.2.2 Baseline

We compare our results to the best previously reported result on BiToD from Lin et al. (2021). This SOTA result was obtained using MinTL (Lin et al., 2020) and using a single mT5-small model to perform all dialogue subtasks.

Contrary to what Lin et al. (2021) reported, we found that mBART-large model outperforms mT5-small in all settings. We have included all the results including MinTL(mBART) in Table 2 for comparison.

6.2.3 Results

The results for our cross-lingual experiment are reported in Table 2. Overall, in the full-shot setting, when training on both source and target language data, we improve the SOTA in JGA by 5.3%, TSR by 3.8%, DSR by 2.9%, API by 2.6%, BLEU by 0.8%, and SER by 2.6%.

Our zero-shot agent achieves 71%, 62%, 40%, and 47% of the performance of a full-shot agent in terms of JGA, TSR, DSR, and BLEU score, respectively. In the 10% few-shot setting, our approach establishes a new SOTA by increasing JGA, TSR, DSR, API, and BLEU absolutely by 13.9%, 15.2%, 14.0%, 15.0%, and 4.8% respectively. Prominently, training with just 10% of the data beats the full-shot baseline which is trained on 100% of the training data, on all metrics except for DSR and BLEU. It also comes within 5% of full training using the Distilled representation on all metrics.

Our Distilled representation improves the performance, especially in few-shot. Comparing our results with that of Lin et al. (2021), in the full-shot monolingual setting (MinTL(mT5) "−Mixed" vs. Ours "−Mixed"), models trained on data with our representation outperform the baseline on all metrics. In the pure few-shot (10%) setting, Ours outperforms MinTL(mT5) significantly in all metrics. This suggests that our Distilled representation and task decomposition are much more effective in low-data settings.

Canonicalization is useful. Comparing "−Translation" with "−Canonicalization", training on canonicalized data significantly improves the results in the zero-shot setting. This is intuitive since canonicalization brings training data closer in vocabulary to the test data in the target language. This improvement comes at almost no cost since translation is done automatically using a dictionary.

Automatic naive translation of the training set does not work for zero-shot. The naive translation approach (i.e. without alignment) completely fails in the zero-shot setting by achieving only 4.7% in TSR, and 1.1% in DSR, as translated entities might no longer match with ones in the annotation. Adding few-shot data helps significantly as the gap closes between "−Alignment" and "−Translation" ablations.

Alignment improves translation quality in all settings and metrics. With alignment, the translation approach performs much better in all settings, establishing a new state-of-the-art in zero and few-shot settings according to almost all metrics. As a general trend, the lower data settings benefit more from alignment. We additionally performed an experiment using the alignment proposed by (Moradshahi et al., 2023). There is a 4.0% drop in TSR and 4.5% in DSR, confirming the benefit of our improved alignment.

Filtering noise for RG improves fluency. We perform an ablation by training separate models on filtered and unfiltered translated agent utterances. The filtering process is described in Section 4.2. In 10% fewshot setting, both BLEU and SER improve by 1.4% confirming that automatically removing poor translations from training data improves the agent response quality. Interestingly, we observe an increase in other metrics too. Since model parameters are shared between all subtasks, enhancing the data quality for one subtask will have a positive impact on the others as well.

7 Conclusion

This paper shows how to build a dialogue agent in a new language automatically, given a dialogue dataset in another language, by using entity-aware machine translation and our new Distilled dialogue representation. The performance can be further improved if a few training examples in the target language are available, and we show that our approach outperforms existing ones in this setting as well.

On the BiToD dataset, our method achieves 3.9% and 2.9% improvement in TSR and DSR, respectively, over the previous SOTA in full-shot setting, and 15.2% and 14.0% in a 10% few-shot setting, showing the effectiveness of our approach. More importantly, training on translated data and only 10% of original training data comes within 5% of full training.

We have implemented our methodology as a toolkit for developing multilingual dialogue agents, which we have released open-source. Our proposed methodology can significantly reduce the cost and time associated with data acquisition for task-oriented dialogue agents in new languages.

8 Limitations

As discussed in Section 2.1, organic (i.e. without the use of translation) multilingual dialogue datasets are scarce, which has limited the scope of our experiments. Our guidelines to improve dialogue representation mentioned in Section 4 are general and applicable to any Human-to-Human or Machine-to-Machine dialogues annotated with slot-values. We have yet to evaluate the generalization of our cross-lingual approach across different languages and datasets, and to Human-to-Human dialogues. For instance, we use a Chinese to English translator in this work. Available translation models for low-resource languages have much lower quality, and this will likely lower the performance of this approach.

Another limitation is the lack of human evaluation for agent responses. BLEU score does not correlate well with human judgment, and SER only accounts for the factuality of the response but not the grammaticality or fluency. This problem is also reported in prior works (see Section 5). Although finding native speaker evaluators for different languages is a challenge (Pavlick et al., 2014), in the future, we wish to address this by conducting human evaluations.

9 Ethical Considerations

We do not foresee any harmful or malicious misuses of the technology developed in this work. The data used to train models is about seeking information about domains like restaurants, hotels and tourist attractions, does not contain any offensive content, and is not unfair or biased against any demographic. This work does focus on two widely-spoken languages, English and Chinese, but we think the cross-lingual approach we proposed can improve future dialogue language technologies for a wider range of languages.

We fine-tune multiple medium-sized (several hundred million parameters) neural networks for our experiments. We took several measures to avoid wasted computation, like performing one run instead of averaging multiple runs (since the numerical difference between different models is large enough), and improving batching and representation that improved training speed, and reduced needed GPU time. Please refer to Appendix 5.2 for more details about the amount of computation used in this paper.

Acknowledgements

This work is supported in part by the National Science Foundation under Grant No. 1900638, the Alfred P. Sloan Foundation under Grant No. G-2020-13938, Microsoft, Stanford HAI and the Verdant Foundation.

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)*.

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.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *arXiv preprint arXiv:1905.12866*.

Wenhu Chen, Jianshu Chen, Yu Su, Xin Wang, Dong Yu, Xifeng Yan, and William Yang Wang. 2018. XL-NBT: A cross-lingual neural belief tracking framework. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 414–424, Brussels, Belgium. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bosheng Ding, Junjie Hu, Lidong Bing, Sharifah Mahani Aljunied, Shafiq Joty, Luo Si, and Chunyan Miao. 2021. Globalwoz: Globalizing multiwoz to develop multilingual task-oriented dialogue systems.

Sauleh Eetemadi and Kristina Toutanova. 2014. Asymmetric features of human generated translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 159–164, Doha, Qatar. Association for Computational Linguistics.

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*.

Savoir faire Linux. 2017. num2words. `https://github.com/savoirfairelinux/num2words`.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374.

Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, Dilek Hakkani-Tür, Jinchao Li, Qi Zhu, Lingxiao Luo, Lars Liden, Kaili Huang, Shahin Shayandeh, Runze Liang, Baolin Peng, Zheng Zhang, Swadheen Shukla, Minlie Huang, Jianfeng Gao, Shikib Mehri, Yulan Feng, Carla Gordon, Seyed Hossein Alavi, David Traum, Maxine Eskenazi, Ahmad Beirami, Eunjoon, Cho, Paul A. Crook, Ankita De, Alborz Geramifard, Satwik Kottur, Seungwhan Moon, Shivani Poddar, and Rajen Subba. 2020. Overview of the ninth dialog system technology challenge: Dstc9.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Chia-Chien Hung, Anne Lauscher, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. 2022. Multi2woz: A robust multilingual dataset and conversational pretraining for task-oriented dialog. *arXiv preprint arXiv:2205.10400*.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021a. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.

Jinchao Li, Qi Zhu, Lingxiao Luo, Lars Liden, Kaili Huang, Shahin Shayandeh, Runze Liang, Baolin Peng, Zheng Zhang, Swadheen Shukla, Ryuichi Takanobu, Minlie Huang, and Jianfeng Gao. 2021b. Multi-domain task-oriented dialog challenge ii at dstc9. In *AAAI-2021 Dialog System Technology Challenge 9 Workshop*.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, Peng Xu, Feijun Jiang, Yuxiang Hu, Chen Shi, and Pascale Fung. 2021. BiToD: A bilingual multi-domain dataset for task-oriented dialogue modeling. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1 pre-proceedings (NeurIPS Datasets and Benchmarks 2021)*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Mehrad Moradshahi, Giovanni Campagna, Sina Semnani, Silei Xu, and Monica Lam. 2020. Localizing open-ontology QA semantic parsers in a day using machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5970–5983, Online. Association for Computational Linguistics.

Mehrad Moradshahi, Victoria Tsai, Giovanni Campagna, and Monica S Lam. 2023. Contextual semantic parsing for multilingual task-oriented dialogues. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. 2014. The language demographics of amazon mechanical turk. *Transactions of the Association for Computational Linguistics*, 2:79–92.

Jun Quan, Shian Zhang, Qian Cao, Zizhong Li, and Deyi Xiong. 2020. RiSAWOZ: A large-scale multi-domain Wizard-of-Oz dataset with rich semantic annotations for task-oriented dialogue modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 930–940, Online. Association for Computational Linguistics.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 432–437.

Evgeniia Razumovskaia, Goran Glavaš, Olga Majewska, Anna Korhonen, and Ivan Vulic. 2021. Crossing the conversational chasm: A primer on multilingual task-oriented dialogue systems. *arXiv preprint arXiv:2104.08570*.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. 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 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

Scrapinghub. 2015. dateparser. `https://github.com/scrapinghub/dateparser`.

Tom Sherborne, Yumo Xu, and Mirella Lapata. 2020. Bootstrapping a crosslingual semantic parser. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 499–517, Online. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. CrossWOZ: A large-scale Chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.

Lei Zuo, Kun Qian, Bowen Yang, and Zhou Yu. 2021. Allwoz: Towards multilingual task-oriented dialog systems for all. *arXiv preprint arXiv:2112.08333*.

A Appendix

A.1 Dialogue Examples

We include the same example from BiToD's English validation set both in our Distilled representation (Table 3) and in the original (Table 4) representation, along with model predictions in the full-shot setting. For brevity, only the first 3 turns are shown.

In Table 4, we observe that the model fails to ask for the hotel price-range in the second turn and makes an API call instead. Since the API call results are carried over between turns in this representation, in the third turn, the model sees those results in the input and falsely assumes it does not need to make an API call anymore, ultimately resulting in an incorrect response. Compare this to our representation in Table 3. This example shows the importance of separation between API call detection and response generation.

Another phenomenon we often observe is that the model asks for more information than it should according to the gold agent dialogue act. As shown in Table 3, in the second turn, the agent requests user to provide the desired location for the hotel as well as the price range. We believe the main reason for this behavior is the randomness in the agent policy of the BiToD's dialogue simulator. For example, if the agent needs to fill out two slots to make an API call, it can do so by requesting both in the same turn, or one turn at a time. This behavior, though reasonable, is penalized during evaluation, and predictions are considered incorrect if they contain extraneous slots.

Turn 1	DST	Input	DST: <state> null <endofstate> <history> USER: I’d like hotel recommendations. <endofhistory>
		Target	(hotels search)
		Prediction	(hotels search)
	ACD	Input	API: <knowledge> null <endofknowledge> <state> (hotels search) <endofstate> <history> USER: I’d like hotel recommendations. <endofhistory>
		Target	no
		Prediction	no
	DAG	Input	ACTS: <knowledge> null <endofknowledge> <state> (hotels search) <endofstate> <history> USER: I’d like hotel recommendations. <endofhistory>
		Target	(hotels search) request rating , request stars
		Prediction	(hotels search) request rating , request stars
	RG	Input	RG: <actions> (hotels search) request rating , request stars <endofactions> <history> USER: I’d like hotel recommendations. <endofhistory>
		Target	Certainly. Do you have any requirements for the hotel’s rating or the number of stars of the hotel?
		Prediction	Do you have a preference on how many stars and what rating the hotel should have?
Turn 2	DST	Input	DST: <state> (hotels search) <endofstate> <history> AGENT_ACTS: (hotels search) request rating , request stars USER: The rating doesn’t matter, but should be at least 5 stars. <endofhistory>
		Target	(hotels search) rating equal_to " don’t care " , stars at_least " 5 "
		Prediction	(hotels search) rating equal_to " don’t care " , stars at_least " 5 "
	ACD	Input	API: <knowledge> null <endofknowledge> <state> (hotels search) rating equal_to " don’t care " , stars at_least " 5 " <endofstate> <history> AGENT_ACTS: (hotels search) request rating , request stars USER: The rating doesn’t matter, but should be at least 5 stars. <endofhistory>
		Target	no
		Prediction	no
	DAG	Input	ACTS: <knowledge> null <endofknowledge> <state> (hotels search) rating equal_to " don’t care " , stars at_least " 5 " <endofstate> <history> AGENT_ACTS: (hotels search) request rating , request stars USER: The rating doesn’t matter, but should be at least 5 stars. <endofhistory>
		Target	(hotels search) request price_level
		Prediction	(hotels search) request location , request price_level
	RG	Input	RG: <actions> (hotels search) request price_level <endofactions> <history> USER: The rating doesn’t matter, but should be at least 5 stars. <endofhistory>
		Target	Do you have a price range for the hotel?
		Prediction	And what about location? Do you have a price range for the hotel?

Turn 3	DST	Input	DST: <state> (hotels search) rating equal_to " don't care " , stars at_least " 5 " <endofstate> <history> AGENT_ACTS_PREV: (hotels search) request rating , request stars AGENT_ACTS: (hotels search) request price_level USER: cheap <endofhistory>
		Target	(hotels search) price_level equal_to " cheap "
		Prediction	(hotels search) price_level equal_to " cheap "
	ACD	Input	API: <knowledge> null <endofknowledge> <state> (hotels search) price_level equal_to " cheap " , rating equal_to " don't care " , stars at_least " 5 " <endofstate> <history> AGENT_ACTS_PREV: (hotels search) request rating , request stars AGENT_ACTS: (hotels search) request price_level USER: cheap <endofhistory>
		Target	yes
		Prediction	yes
	DAG	Input	ACTS: <knowledge> (hotels search) available_options " 4 " , location " Mong Kok | Kowloon | Yau Tsim Mong District " , name " Royal Plaza Hotel " , price_level " cheap " , price_per_night " 793 HKD " , rating " 9 " , stars " 5 " <endofknowledge> <state> (hotels search) price_level equal_to " cheap " , rating equal_to " don't care " , stars at_least " 5 " <endofstate> <history> AGENT_ACTS_PREV: (hotels search) request rating , request stars AGENT_ACTS: (hotels search) request price_level USER: cheap <endofhistory>
		Target	(hotels search) offer available_options equal_to " 4 " , offer name equal_to " Royal Plaza Hotel " , offer rating equal_to " 9 "
		Prediction	(hotels search) offer available_options equal_to " 4 " , offer name equal_to " Royal Plaza Hotel " , offer rating equal_to " 9 "
	RG	Input	RG: <actions> (hotels search) offer available_options equal_to " 4 " , offer name equal_to " Royal Plaza Hotel " , offer rating equal_to " 9 " <endofactions> <history> USER: cheap <endofhistory>
		Target	Okay. There are 4 hotels available. I recommend the Royal Plaza Hotel, which has a 9 rating.
		Prediction	There are 4 available hotels. I recommend Royal Plaza Hotel. Its rating is 9.

Table 3: An example from BiToD English validation set in Distilled representation, along with our mBART model predictions. For brevity, only the first 3 turns are shown.

Turn 1	DST	Input	Track Dialogue State:<knowledge><dialogue_state> <user> I'd like hotel recommendations.
		Target	<API> hotels search
		Prediction	<API> hotels search
	API/ **Response**	Input	Generate Response:<knowledge><dialogue_state> <API> hotels search<user> I'd like hotel recommendations.
		Target	Certainly. Do you have any requirements for the hotel's rating or the number of stars of the hotel?
		Prediction	What rating would you like the hotel to have?
Turn 2	DST	Input	Track Dialogue State:<knowledge><dialogue_state> <API> hotels search<user> I'd like hotel recommendations.<system> Certainly. Do you have any requirements for the hotel's rating or the number of stars of the hotel?<user> The rating doesn't matter, but should be at least 5 stars.
		Target	<API> hotels search<slot> rating<relation> equal_to<value> don't care<slot> stars<relation> at_least<value> 5
		Prediction	<API> hotels search<slot> rating<relation> equal_to<value> don't care<slot> stars<relation> at_least<value> 5
	API/ **Response**	Input	Generate Response:<knowledge><dialogue_state> <API> hotels search<slot> rating<relation> equal_to<value> don't care<slot> stars<relation> at_least<value> 5<user> I'd like hotel recommendations.<system> Certainly. Do you have any requirements for the hotel's rating or the number of stars of the hotel?<user> The rating doesn't matter, but should be at least 5 stars.
		Target	Do you have a price range for the hotel?
		Prediction	I found 43 hotels. I recommend Cordis, Hong Kong, which has a rating of 10.
Turn 3	DST	Input	Track Dialogue State:<knowledge><dialogue_state> <API> hotels search<slot> rating<relation> equal_to<value> don't care<slot> stars<relation> at_least<value> 5<user> The rating doesn't matter, but should be at least 5 stars.<system> Do you have a price range for the hotel?<user> cheap
		Target	<API> hotels search<slot> price_level<relation> equal_to<value> cheap
		Prediction	<API> hotels search<slot> price_level<relation> equal_to<value> cheap
	API/ Response	Input	Generate Response:<knowledge><dialogue_state> <API> hotels search<slot> rating<relation> equal_to<value> don't care<slot> stars<relation> at_least<value> 5<slot> price_level<relation> equal_to<value> cheap<user> The rating doesn't matter, but should be at least 5 stars.<system> Do you have a price range for the hotel?<user> cheap
		Target	<API> hotels search
		Prediction	–
	API/ **Response**	Input	Generate Response:<knowledge> [hotels]<slot> name<value> Royal Plaza Hotel<slot> location<value> Mong Kok<value> Kowloon<value> Yau Tsim Mong District<slot> price_level<value> cheap<slot> price_per_night<value> 793 HKD<slot> rating<value> 9<slot> stars<value> 5<slot> available_options<value> 4<dialogue_state> <API> hotels search<slot> rating<relation> equal_to<value> don't care<slot> stars<relation> at_least<value> 5<slot> price_level<relation> equal_to<value> cheap<user> The rating doesn't matter, but should be at least 5 stars.<system> Do you have a price range for the hotel?<user> cheap<API> hotels search
		Target	Okay. There are 4 hotels available. I recommend the Royal Plaza Hotel, which has a 9 rating.
		Prediction	The hotel costs 839 HKD per night.

Table 4: Same example as in Table 3 but in the original representation from Lin et al. (2021), along with MinTL(mT5) model predictions. For brevity, only the first 3 turns are shown.