Multi-Domain Task Completion Dialog Challenge II at DSTC9

**Jinchao Li*[1], Qi Zhu*[2], Lingxiao Luo[2], Lars Liden[1], Kaili Huang[2],
Shahin Shayandeh[1], Runze Liang[2], Baolin Peng[1], Zheng Zhang[2],
Swadheen Shukla[1], Ryuichi Takanobu[2], Minlie Huang[2], Jianfeng Gao[1]**

[1]Microsoft Research, Redmond, WA, US
{jincli, laliden, shahins, bapeng, swads, jfgao}@microsoft.com
[2]Tsinghua University, Beijing, China
{zhu-q18, luolx17, hkl16, liangrz20, z-zhang15, gxly19}@mails.tsinghua.edu.cn aihuang@tsinghua.edu.cn

Abstract

The paper provides an overview of the "Multi-Domain Task-Completion Dialog Challenge II" track at the 9th Dialog System Technology Challenge (DSTC9). Two tasks are introduced in this track. One is end-to-end multi-domain task-completion, which focuses on building end-to-end task completion dialog systems. The other is cross-lingual dialog state tracking, which seeks to build a tracker for the target language using the source language resources. We describe the task settings, baselines, evaluation methods, and submission results for both tasks.

Introduction

As part of DSTC9 (Gunasekara et al. 2020), in this track, we foster the progress of building multi-domain task-oriented dialog systems in two aspects: dialog complexity and adaptation to new languages. One task is the end-to-end task-oriented dialog task aiming to solve the complexity of building end-to-end dialog systems. The other is cross-lingual dialog state tracking (DST) to address the language adaption problem in the DST task. In the next sections, we discuss the setup, evaluation, baseline, and result of the end-to-end task completion task and the cross-lingual DST task, respectively.

End-to-End Task-Completion Task

In DSTC8 (Kim et al. 2019), we employed CONVLAB (Lee et al. 2019) and proposed End-to-End Multi-Domain Task-Completion Task in *Multi-Domain Task-Completion Dialog Challenge* Track, where participants were encouraged to explore all possible approaches to build an end-to-end task-oriented dialog system that takes natural language as input and generates natural language response as output based on the MultiWOZ 2.0 dataset. The evaluation result of the challenge (Li et al. 2020) and empirical analysis of models in CONVLAB (Takanobu et al. 2020) demonstrate that rule-based pipeline systems can outperform state-of-the-art machine learning models, and the performance of component-wise models is not always consistent with the performance of an entire system using the end-to-end evaluation. These findings explain the landscape of most dialog development technology stacks in the industry while raising the challenge of developing more effective dialog systems using machine learning models. Interestingly, the winning team at the human evaluation phase built their model by leveraging GPT-2 (Radford et al. 2019; Ham et al. 2020), and significantly outperforms other teams in terms of success rate, understanding score, and response score. Meanwhile, with similar model training paradigms based on GPT-2, SOLOIST (Peng et al. 2020) and SIMPLETOD (Hosseini-Asl et al. 2020) also achieved top performance in the MultiWOZ leaderboard. Readers can refer to (Gao et al. 2020) for an overview of this type of dialog development approach.

The task this year is a direct extension of the one in DSTC8. We consider the same settings with some major changes in the dataset version and evaluation approaches. We also provide the latest development platform CONVLAB-2 (Zhu et al. 2020b) to mitigate the efforts of developing and evaluating dialog systems. Participants are encouraged to explore all possible approaches with no restriction on dialog system architectures.

Resources

Dataset Participants are expected to build dialog systems based on MultiWOZ 2.1 this year as opposed to MultiWOZ 2.0 in DSTC8. Compared with MultiWOZ 2.0, MultiWOZ 2.1 re-annotated states and utterances based on the original utterance to fix the original noisy annotation. It also contains user dialog act annotation, which is missing in MultiWOZ 2.0. Meanwhile, we also provide a label corrected version of MultiWOZ 2.1, which addresses issues including entity matching, address splitting, inconsistent labeling, and missing spans. This label adjusted dataset was released at the challenge website [1] for participants' reference. Despite the fact that the evaluation is based on MultiWOZ 2.1, participants are allowed to train their models using any dataset or pre-trained model.

*Equal contribution.

[1] https://github.com/ConvLab/ConvLab-2/blob/master/data/multiwoz/MultiWOZ2.1

ConvLab-2 ConvLab-2 is a dialog development platform built based on ConvLab. It inherits the framework and models from ConvLab and incorporates new features, including the latest state-of-the-art models and tools for evaluation and diagnosis. In this challenge, ConvLab-2 mainly serves as the following two functionalities:

- Dialog System Development. ConvLab-2 consists of a wide range of models for natural language understanding (NLU), dialog state tracker(DST), policy learning, natural language generation (NLG), and end-to-end models. These models are readily trained using MultiWOZ 2.1 and integrated with the database so participants can build an end-to-end dialog system with ease.
- Dialog System Evaluation. ConvLab-2 contains the latest tools for automatic evaluation and human evaluation using MultiWOZ 2.1 so that participants can smoothly run offline evaluation. It also provides rich statistics extracted from the conversations between the user simulator and the dialog system for diagnosis purposes.

Baseline The baseline model was generated using a hierarchical set of Hybrid Code Networks (HCN), where each HCN consists of an LSTM as described in (Williams, Atui, and Zweig 2017). Every user utterance is presented to the top of the hierarchy (the "dispatch" level), where the HCN was trained to identify which domain(s) are relevant to the incoming user utterance given the state of the conversation. The utterance is then passed on to one or more HCN domain models for the identified domains. At the second level of the hierarchy (the "domain" level), the HCN model was trained to identify one or more dialog act categories relevant to the incoming user utterance (i.e., "hotel-inform"). Subsequent tiers (up to two per domain) were trained to select the appropriate dialog act for the given utterance (i.e., "hotel-inform-parking"). The selected dialog acts are then propagated back up the hierarchy and assembled at the dispatch level. Entity labels are also passed down and back up the hierarchy as appropriate for each domain. The total model was comprised of 76 individual HCN models, where each model only receives the portion of the conversation relevant to that model.

Evaluation

Automatic Evaluation We employ the user-simulator in ConvLab-2 for automatic evaluation with details listed below:

- User Simulator: The user-simulator is constructed by assembling a BERT-based natural language understanding (NLU) model (Devlin et al. 2019) , an agenda-based user simulator (Schatzmann et al. 2007) and a rule-based natural language generation (NLG) module. It is similar to the user simulator used in DSTC8 except that MILU model is replaced with a BERT-based model for the NLU module.
- Goal Sampling: We first calculate the frequency of all slot combinations in the MultiWOZ dataset and then sample the user goal based on the slot combinations' distribution. Each slot value in the user goal is sampled from the database with an additional mechanism enforced to guarantee that at least one entity in the database meets the full constraints. Compared with DSTC8, the slot sampling strategy is shifted from individual slot sampling to slot combination sampling. Since slot combinations are directly extracted from the MultiWOZ dataset, this change makes the domain/slot distribution and simulated dialogs more consistent with the original dataset.
- Evaluation Metrics: We report a range of metrics, including dialog success rate, number of turns, book rate, complete rate, and precision/recall/F1 score for slot detection. One primary difference with last year is the calculation of the success rate. In DSTC8, a dialog is successful if the recall score for slot detection and book rate is 1. The recall score for slot detection only cares whether all requested slots are filled (with some additional logic to check input format). This year, we add database grounding constraints. After collecting the values of all requested slots in the conversation, the evaluator creates queries based on inform/request slot values and search the database to confirm whether at least one entity exists in the database. Meanwhile, to mitigate potential NLU errors in the user simulator, we also apply fuzzy matching to requested slot values at the database query stage.
- Number of Dialogs: The number of dialogs for automatic evaluation is increased to 1000, as opposed to 500, for each submission.

Human Evaluation In human evaluation, we host the submitted dialog systems as bot services and allow Amazon Mechanic Turkers to communicate with the bots via natural language. The MTurkers will provide scores based on language understanding correctness and response appropriateness on a 5 point Likert-scale and judge whether the dialog is successful. Compared with DSTC8, there are two primary changes.

- Success Rate: Based on the original design of our human evaluation toolkits, MTurkers do not have access to the database. They have no clue whether the provided slot values are valid, so the success judgment is subjective (success rate without database grounding). This year, we add additional metrics to handle the database grounding issue by adding extra steps in human evaluation. Once MTurkers mark a dialog as successful, they are also asked to provide all requested slot values for database query verification purposes. We then report the success rate with grounding after verifying whether the requested slot values match a database record at the post-processing stage. The average value of success rate with grounding and without grounding is taken for the final ranking.
- Number of Dialogs: In DSTC8, we ran 100 conversations for each system. For teams with a very similar success rate, we increase the number of conversations until we ensure the relative ranking is stable. This year, we significantly increased the number of dialogs up to 500 for each team.

Submissions

As per our submission policy, each team can submit up to 5 models, with the best model considered for the final ranking. At the final submission stage, we have received 34 models from 10 teams. The automatic evaluation result is shown in Table 1. We then filtered out low-performance models based on the automatic evaluation result while maintaining each team's best model. Out of 34 models, 21 models were evaluated in the human evaluation phase, with the best-performing models shown in Table 2. The top 7 teams were evaluated using 500 dialogs and the remaining 3 teams 200 - 250 dialogs. Below is a list of dialog system descriptions for each team based on the model description files and code from the submissions.

- **Team 1:** This is an end-to-end dialog system constructed based on the pre-trained dialog generation model PLATO-2 (Bao et al. 2020). Given the dialog context, this model generates the dialog state, system action, and system response simultaneously. The dialog state is used as the constraint for database query, and the system action is then refreshed according to the queried results. If there is an update in the system action, the model will re-generate the final system response.
- **Team 2:** The system is a hybrid end-to-end neural model that consists of a pre-train & fine-tune architecture based on GPT-2, a fault tolerance mechanism to correct errors, and various pre/post-processing modules for model generalization improvement. The strategy of pre-training and fine-tuning is borrowed from SOLOIST (Peng et al. 2020) and Gururangan et al.'s work (Gururangan et al. 2020). Both domain adaptive (using GPT-2 objectives) and task adaptive (using task-specific objectives) pre-training are applied on domain-related datasets before fine-tuning on the MultiWOZ dataset. The fault tolerance mechanism adjusts the GPT-2 decoder to produce different but potentially correct outputs when errors or inappropriate responses occur. The pre-processing module normalizes dialog slots and delexicalizes utterances, and the post-processing module recovers the delexicalization using rules.
- **Team 3:** This team trains a GPT based model on delexicalized data and adds post-processing stages to enhance the performance.
- **Team 4:** This team builds their end-to-end model based on SIMPLETOD (Hosseini-Asl et al. 2020), and leverages BERT model for NLU part to replace generated belief states that the models are prone to make errors.
- **Team 5:** This team takes Ham et al.'s work (Ham et al. 2020) as the primary reference but uses a different sample and delexicalization strategy. They train their GPT-2 based model using two subtasks: 1. next token sequence prediction, which consists of context history, domain-specific slot constraints, system dialog act, and system delexicalized response. 2. prediction of the consistency of the system delexicalized response and other sequence components mentioned above.

Table 1: Automatic Evaluation Result (Best Submissions)

Team	SR	CR	BR	Inform P/R/F1	Turn S/A
1	**93**	95.2	94.6	84.1/96.2/88.1	12.5/12.7
2	91.4	96.9	96.2	80.2/97.3/86.0	15.3/15.7
3	90.8	94.4	96.7	81.0/95.4/85.9	13.4/13.6
4	89.8	94.6	96.3	72.4/96.0/80.1	15.1/15.8
5	83.3	88.5	89.1	81.1/90.3/83.5	13.5/13.8
6	67.7	88.5	90.8	70.4/85.6/75.2	12.8/14.2
7	57.8	87.1	85	68.7/81.6/72.6	13.7/16.4
8	52.6	66.9	66.7	57.5/80.7/64.8	13.2/22.5
9	44.4	50	26.5	57.9/64.5/58.9	12.2/14.6
10	21.4	40.7	0	55.4/60.0/54.1	11.0/25.9
BS	85	92.4	91.4	79.3/94.9/84.5	13.8/14.9

BS: Baseline, SR: Success Rate, CR: Complete Rate, BR: Book Rate, Inform P/R/F1: Prec./Recall/F1 score of slots prediction, Turn S/A: Turns for successful and all dialogs, respectively.

Table 2: Human Evaluation Result (Best Submissions)

Team	SRa	SRwg	SRog	Under.	Appr.	Turn	Rank
1	**74.8**	70.2	79.4	4.54	4.47	18.5	1
2	**74.8**	68.8	80.8	4.51	4.45	19.4	1
7	72.3	62	82.6	4.53	4.41	17.1	3
6	70.6	60.8	80.4	4.41	4.41	20.1	4
3	67.8	60	75.6	4.56	4.42	21	5
4	60.3	51.4	69.2	4.49	4.22	17.7	6
5	58.4	50.4	66.4	4.15	4.06	19.7	7
9	55.2	43.2	67.2	4.15	3.98	19.2	8
8	35	26	44	3.27	3.15	18.5	9
10	19.5	6	33	3.23	2.93	18.8	10
BS	69.6	56.8	82.4	4.34	4.18	18.5	N/A

SRa: average success rate, SRwg: success rate w/ grounding, SRog: success rate w/o grounding, Under.: understanding score, Appr.: appropriateness score, BS: Baseline.

- **Team 6:** This is a pipeline system based on BERT NLU, GRU-based DST, GRU-based policy, and GRU-based NLG. BERT NLU only takes the utterance at the current turn as the input. The NLU result is combined with the previous belief state (domains and slot-values) to predict the new belief state, which is then used to generate the new system action. Finally, the generated system action and previous input are fed to NLG for system response generation.
- **Team 7:** This team uses GPT-2 as the backbone architecture for the dialog system. Similarly to Sequicity (Lei et al. 2018) or GRU (Peng et al. 2020), the language model is used first to generate the belief state in a fixed format and then to generate a final delexicalized response.
- **Team 8:** This system is based on a BERT NLU model, a rule-based DST model, and a word policy model MARCO (Wang et al. 2020a).
- **Team 9:** This system is built on a transformer-based pre-trained model.
- **Team 10:** This team builds the end-to-end model based on GPT-2 model.

Results and Discussions

Team 1 reaches the best success rate of 93% in automatic evaluation and the best average success rate of 74.8% in human evaluation. Team 2, while ranks 2nd in the automatic evaluation, achieves the same average success rate as Team 1 in human evaluation. Both of them built their dialog systems with an end-to-end modeling approach by leveraging transformer-based models, with Team 1 using PLATO-2 and Team 2 using GPT-2. The success of transformer-based end-to-end modeling is consistent with the results in DSTC8 human evaluation and the MultiWOZ leaderboard [2]. As one of the primary differences from DSTC8, we consider the success rate with database grounding this year. Team 1 handles the grounding problem exceptionally well and remains the best team in SRwg, with only 9.2% drop from SRog (success rate without grounding) to SRwg (success rate with grounding) in human evaluation. Some teams suffer more success rate drops than others due to database grounding despite using similar modeling architectures.

When comparing automatic evaluation in Table 1 and human evaluation in Table 2, the rankings of most teams are relatively stable except Team 6 and Team 7. These two teams rank 6 and 7 in automatic evaluation, respectively, but rank 4 and 3 in human evaluation. The ranking discrepancy can be partially explained by the fact that both teams do not handle the grounding problem well (both have high SRog but moderate SRwg), but it is still unclear why the gap is huge.

In DSTC8, out of 11 teams, one team uses transformer/GPT-2 based end-to-end models, one team uses word DST + word policy, and all the rest 9 teams use component-wise models (most of them used rules for some components). This year, 8 out 10 teams use transformer-based models to build an end-to-end neural network with a shared transformer-structure for dialog state, system action, and response prediction. This indicates that there is a clear trend of shifting from building dialogs by assembling component-wise modules to end-to-end learning using transformer-based models, and that transformer-based models start to dominate the leaderboard other than rule-based systems.

By comparing the human evaluation result between DSTC8 and DSTC9 [3], we can see a significant improvement in dialog development technology over the past year. The best team in DSTC8 achieves 68.3% success rate without grounding (DSTC8 only considers success rate without database grounding), but it will only rank 7 in the leaderboard this year, where the top-performing team reaches 82.6%.

Cross-Lingual Dialog State Tracking Task

With the rapid globalization process, the need for adapting dialog systems in rich-resource languages to low-resource languages is increasing. To verify the language portability of existing monolingual technologies and advance the state-of-the-art cross-lingual technologies in building dialog systems, we introduce the task of cross-lingual dialog state tracking in this track.

Given the context, the dialog state tracking (DST) module predicts the dialog state that summarizes user constraints until the current turn. Dialog state can be used to fetch relative information from a database, making DST one of the critical components in building a dialog system. In DSTC5 (Kim et al. 2016), a cross-language dialog state tracking task was introduced, requiring the participants to build a tracker for the target language using resources in the source language and the corresponding machine-translated sentences in the target language.

In this task, following a similar scheme as in DSTC5, our goal is to build a cross-lingual dialog state tracker with a training set in the rich-resource language and a small development set in the low-resource language. We offer two sub-tasks: 1) cross-lingual transfer from English to Chinese using MultiWOZ 2.1 (Eric et al. 2019) dataset and 2) cross-lingual transfer from Chinese to English using CrossWOZ (Zhu et al. 2020a) dataset. For each sub-task, we additionally provided machine translations of the original dataset and ontology. We collected new dialogs in the target language as a test set for evaluation.

Resources

Compared with previous work (Kim et al. 2016; Mrksic et al. 2017; Schuster et al. 2019), we employ newly proposed much larger multi-domain datasets MultiWOZ 2.1 and CrossWOZ. For each sub-task, we prepared the data following the same process: 1) collect 500 new dialogs in the source language, 2) translate the ontology to the target language, 3) translate the dialogs and annotations of both the original dataset and the new dataset. We sampled 250 dialogs from the original dataset as a development set. Translated new dataset serves as a test set. We released 250 dialogs sampled from the test set without any annotation as a public test set and reserved the other 250 dialogs as a private test set. Statistics of collected data is shown in Table 3.

Table 3: Statistics of collected data in the target language. The training set is translated by Google Translate. The development set and test set are first translated by Google Translate and then corrected by humans.

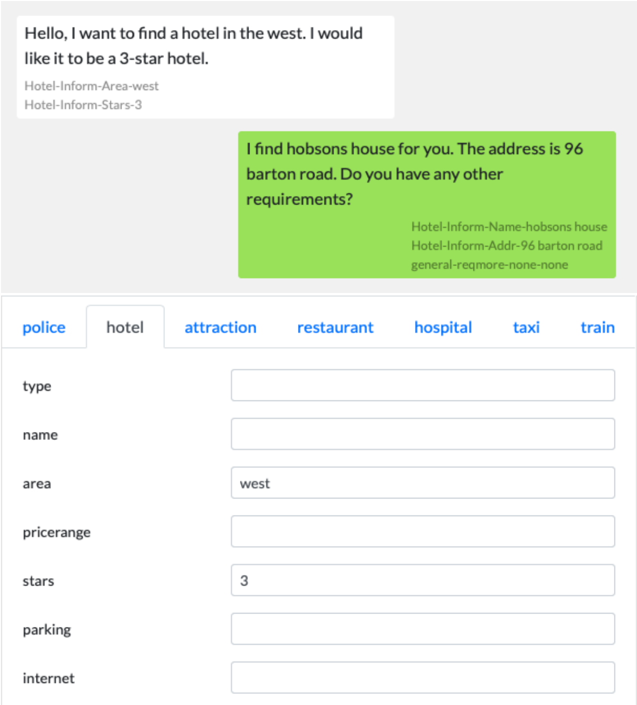
	MultiWOZ ZH			CrossWOZ EN		
	Train	Dev	Test	Train	Dev	Test
# Dialogs	10433	250	500	6012	250	500
# Utterances	142974	3646	5788	101626	4188	7604
Avg. domains	1.83	1.92	1.90	3.25	3.26	3.30
Avg. utterances	13.70	14.58	11.58	16.90	16.75	15.21
Avg. tokens	13.97	14.77	11.31	17.45	17.99	20.68
# Slots		30			26	
# Values		1971			8206	

Original Datasets Two large scale multi-domain task-oriented dialog datasets, MultiWOZ 2.1 and CrossWOZ, are used for en→zh and zh→en respectively. MultiWOZ 2.1 contains over 10,000 dialogs spanning 7 domains, while CrossWOZ contains over 6,000 dialogs spanning 5 domains.

[2] https://github.com/budzianowski/multiwoz

[3] https://convlab.github.io/

Test Data Collection For each sub-task, we collected 500 new dialogs in the source language. We first generated new user goals in natural language using the goal generator from ConvLab-2. Then we collected the dialogs in a similar way as described in the CrossWOZ paper. We adapted the data collection website of CrossWOZ, which allows two workers to converse synchronously and make annotations online. Following the Wizard-of-Oz setting, one worker acts as the user seeking information according to the user goal, while the other acts as the wizard that can access the database to provide services. During the conversation, both sides need to annotate the dialog acts of their utterances, and the wizard should additionally log down the dialog states that are used as queries over the database. An example is shown in Figure 1. Before the formal data collection, we trained the workers by asking them to complete a small number of dialogs and giving them feedback. In total, 66 and 50 workers participated in MultiWOZ and CrossWOZ data collection, respectively.



Hello, I want to find a hotel in the west. I would like it to be a 3-star hotel.

Hotel-Inform-Area-west
Hotel-Inform-Stars-3

I find hobsons house for you. The address is 96 barton road. Do you have any other requirements?

Hotel-Inform-Name-hobsons house
Hotel-Inform-Addr-96 barton road
general-reqmore-none-none

type	
name	
area	west
pricerange	
stars	3
parking	
internet	

Figure 1: Screenshots of the data collection website. Top: conversation. Bottom: dialog state. Users need to annotate dialog acts (below the utterance), while systems need to annotate dialog acts and dialog states.

Ontology Translation To ensure the consistency of the translations of dialogs and corresponding annotations, we constructed an ontology dictionary. We extracted the ontology from dialog act and dialog state annotations of both the original and the test datasets and used Google Translate to translate them to the target language. Then we employed human translators to correct the translations for some slots that may not be faithfully translated, such as "name" and "address". Combining the ontology dictionary with the machine translator for handling OOV values, we provided a function that can translate dialog act and dialog state annotations. We also translated the database using this function.

Dialog Translation We used Google Translate and ontology dictionary to translate the original dataset and the test set. Before translating a dialog, we replaced the values that appeared in the dialog with their translations in the dictionary. This process is vital to ensure the translation consistency of the same values in different contexts. We sampled 250 dialogs from the original dataset as the development set. Human translators were employed to proofread the translations of the development and the test set.

Baseline We adapted SUMBT (Lee, Lee, and Kim 2019) as the baseline model for both sub-tasks. SUMBT uses BERT to encode the system and user utterance in the current turn. For each slot, another pre-trained and fixed BERT encodes a phrase of the domain and slot words (e.g., "restaurant – price range") as its representation. This representation is used to query the utterances representations through multi-head attention. Each slot's values defined in the ontology are also encoded using the pre-trained and fixed BERT. To predict the value of a slot at each turn, an RNN collects the slot-conditioned representation of the context and retrieves the value that has the closest representation to this representation among all possible values. We used the translated training set of the original dataset to train SUMBT, which is the "Translate-Train" setting in cross-lingual transfer learning. For MultiWOZ (en→zh) sub-task, we used Chinese pre-trained BERT[4] (Cui et al. 2019).

Evaluation

We evaluate the performance of the dialog state tracker using the following metrics:

- Joint Goal Accuracy. This metric evaluates whether the predicted dialog state is exactly equal to the ground truth.
- Slot Accuracy. This metric evaluates whether the predicted label for each individual slot is exactly equal to the ground truth, averaged over all slots.
- Slot Precision/Recall/F1. Since the slot accuracy may be dominated by the situation that both the prediction and label are empty, we use these metrics to evaluate the prediction for non-empty slots only, micro-averaged over all slots. We show the difference between these metrics and slot accuracy in Table 4.

Each submission contains the predictions for the public test set and the model used to make predictions for the private test set. The results are averaged over the public and private test set. The final ranking is solely based on the joint goal accuracy.

[4] https://huggingface.co/hfl/chinese-bert-wwm-ext

Table 4: Calculate slot accuracy and slot precision/recall/f1 for each slot. **acc**: count for accuracy. **TP**: true positive. **FP**: false positive. **FN**: false negative.

Label Pred	empty	non-empty
empty	acc+=1	FN+=1

non-empty	FP+=1	if pred = label:
		TP+=1, acc+=1
		else:
		FN+=1, FP+=1

Submissions

For each sub-task, one team is allowed to submit up to 5 models, and the best model is used for the final ranking. At the final submission stage, we have received 10 models for MultiWOZ (en→zh) and 8 models for CrossWOZ (zh→en) from the same 3 teams.

- **Team 1:** They used modified CHAN model (Shan et al. 2020) for both MultiWOZ and CrossWOZ sub-tasks. Inspired by SOM-DST (Kim et al. 2020), they incorporated a four-class state operation (i.e., update, delete, carryover, dontcare) prediction task into the CHAN model. They also modified the labels of original datasets for this auxiliary task. Besides the provided data translated by Google Translate, they used the data translated by their own translation model for training.
- **Team 2:** For both sub-tasks, their best model was based on SOM-DST. They used different role symbols to obtain information from system-agent and user-agent to distinguish the recommendations from system-agent and intents from user-agent. Since the generated values may have some discrepancies with ground-truth labels, they used ontology and some handcraft rules to post-process. They used some approaches to augment the training data, which improved performance. Similar to Team 1, they used Baidu Translate to translate the original dataset into the target language and mix them with provided translations as the training data. According to the bad case analysis, they augmented some single-domain dialogue sessions from multi-domain dialogue sessions and replaced some slot values with the ones which models performed poorly. They also tried TripPy (Heck et al. 2020), which makes use of three copy mechanisms to fill slots with values. Since the model needs system side dialog acts which are not available in the test set, they used ontology and synonyms to extract dialog action for some specified slots, such as "name". They have tried some cross-lingual pretrained models such as XLM (Lample and Conneau 2019) and trained the Chinese and English datasets simultaneously. However, the models did not benefit from the cross-lingual data according to the result of their experiment.
- **Team 3:** They formulated the dialog state tracking as a sequence generation problem. The models take dialog history as input and output pairs of slot names and slot values. Their best model used mBART (Liu et al. 2020) trained on the machine translations of the original datasets for both sub-tasks. They also tried GPT-2 (Radford et al. 2018) and CDialGPT2$_{LCCC-base}$ (Wang et al. 2020b) for translated CrossWOZ and MultiWOZ datasets respectively but got worse performance than mBART. Since mBART can take data in multiple languages as input, they tried to use the original dataset in the source language, and further use the data from the other sub-task for training. However, both strategies gave worse results.

Results

The results of MultiWOZ (en→zh) and CrossWOZ (zh→en) sub-tasks are shown in Table 5 and 6 respectively. During the evaluation, we found that the newly collected CrossWOZ data miss a number of the "name" labels when the user accepts the attraction/hotel/restaurant recommended by the system. Therefore, we utilized the database search results, which are selected by the system to compose the response, to correct empty "name" labels using handcraft rules[5]. We also provide an updated leaderboard for CrossWOZ in Table 7. This change is considered as applying two different evaluation approaches, and both of the original and new leaderboards are valid. Nevertheless, the new leaderboard is preferred.

For MultiWOZ (en→zh), Team 1 achieves the best joint goal accuracy of 62.37%, and Team 2 gets a slightly lower score of 62.08%. The performance of our baseline model is 55.56%. Compared with the results on MultiWOZ 2.1 English leaderboard, these numbers are much better, which can be attributed to the difference between our test set and the original one, as reflected in the dialog length and utterance length shown in Table 3.

For CrossWOZ (zh→en), Team 2 reaches a much better performance (32.30%) than other teams on the updated leaderboard. Our baseline model only gets 13.02% joint goal accuracy. Compared with MultiWOZ, the joint goal accuracy is much lower, possibly because CrossWOZ dataset is more difficult with a much larger value set as shown in Table 3.

Table 5: MultiWOZ Leaderboard. The results are from the best submissions from each team.

Team	JGA	SA	Slot P/R/F1	JGA(pub/pri)	Rank
1	62.37	98.09	92.15/94.02/93.07	62.70/62.03	1
2	62.08	98.10	90.61/96.20/93.32	63.25/60.91	2
3	30.13	94.40	87.07/74.67/80.40	30.53/29.72	3
BS	55.56	97.68	92.02/91.10/91.56	55.81/55.31	N/A

JGA: joint goal accuracy, SA: slot accuracy, Slot P/R/F1: slot precision/recall/f1, pub/pri: public/private test set.

Error Analysis for Baseline

The baseline model achieves a relatively high joint goal accuracy on MultiWOZ but a much lower score on CrossWOZ, just like the participants' submissions. To analyze errors made by the baseline model, we calculate each slot's error rate (i.e., $1 - a$, where a is the accuracy for that slot). Figure 2 plots normalized error rates and nonempty rates (i.e.,

[5] https://github.com/thu-coai/ConvLab-2/blob/master/convlab2/dst/dstc9/utils.py#L13-L68

Table 6: CrossWOZ Leaderboard. The results are from the best submissions from each team.

Team	JGA	SA	Slot P/R/F1	JGA(pub/pri)	Rank
3	16.86	89.11	68.26/62.85/65.45	16.82/16.89	1
1	15.28	90.37	65.94/78.87/71.82	15.19/15.37	2
2	13.99	91.92	72.63/78.90/75.64	14.41/13.58	3
BS	7.21	85.13	55.27/46.15/50.30	7.41/7.00	N/A

Table 7: CrossWOZ Leaderboard (Updated Evaluation). The results are from the best submissions from each team.

Team	JGA	SA	Slot P/R/F1	JGA(pub/pri)	Rank
2	32.30	94.35	81.39/82.25/81.82	32.70/31.89	1
1	23.96	92.94	74.96/83.41/78.96	23.45/24.47	2
3	15.31	89.70	74.78/64.06/69.01	14.25/16.37	3
BS	13.02	87.97	67.18/52.18/58.74	13.30/12.74	N/A

the ratio of values of a slot that is not empty) for all slots. A relatively high error rate and a lower nonempty rate mean a slot is difficult. We can observe that the model performs poor in “name”, “Hotel-type/stay/parking”, and “Taxi-destination/departure” slots for MultiWOZ, and “Attraction-duration”, “Restaurant-dishes/rating”, “Hotel-Hotel Facilities” and “nearby attract./rest./hotel” for CrossWOZ. Some of these slots have lower error rates but also lower nonempty rates. The baseline model may be incapable of overcoming the data sparsity of these slots.

Discussion

To our surprise, all the best models are trained on monolingual machine translated data instead of both the original data and translations. Although “Translate-Train” is a strong setting in cross-lingual transfer learning, Huang et al. (2019) found that fine-tuning a multilingual pre-trained model on the original data and its translations in multiple languages together is more powerful. However, Team 2 and 3 got negative results when trained XLM/mBART on the Chinese and English datasets simultaneously. The performance of “Translate-Train” partially depends on the machine translator, which may be why Team 1 and 2 augmented the data using another translator to translate the original dataset. Team 1 and 2 modified DST models that are state-of-the-art on English MultiWOZ 2.1 dataset and got strong performance on Chinese MultiWOZ 2.1, verifying these models’ language portability.

Conclusion

In this paper, we summarized the end-to-end task-completion task and the cross-lingual dialog state tracking task at DSTC9. The end-to-end task-completion task is a continuation of last year and requires participants to build an end-to-end dialog system. With a year, there is a clear trend of shifting from using rule-based and component-wise models to transformer-based end-to-end modeling approaches for dialog system development. All top teams employed

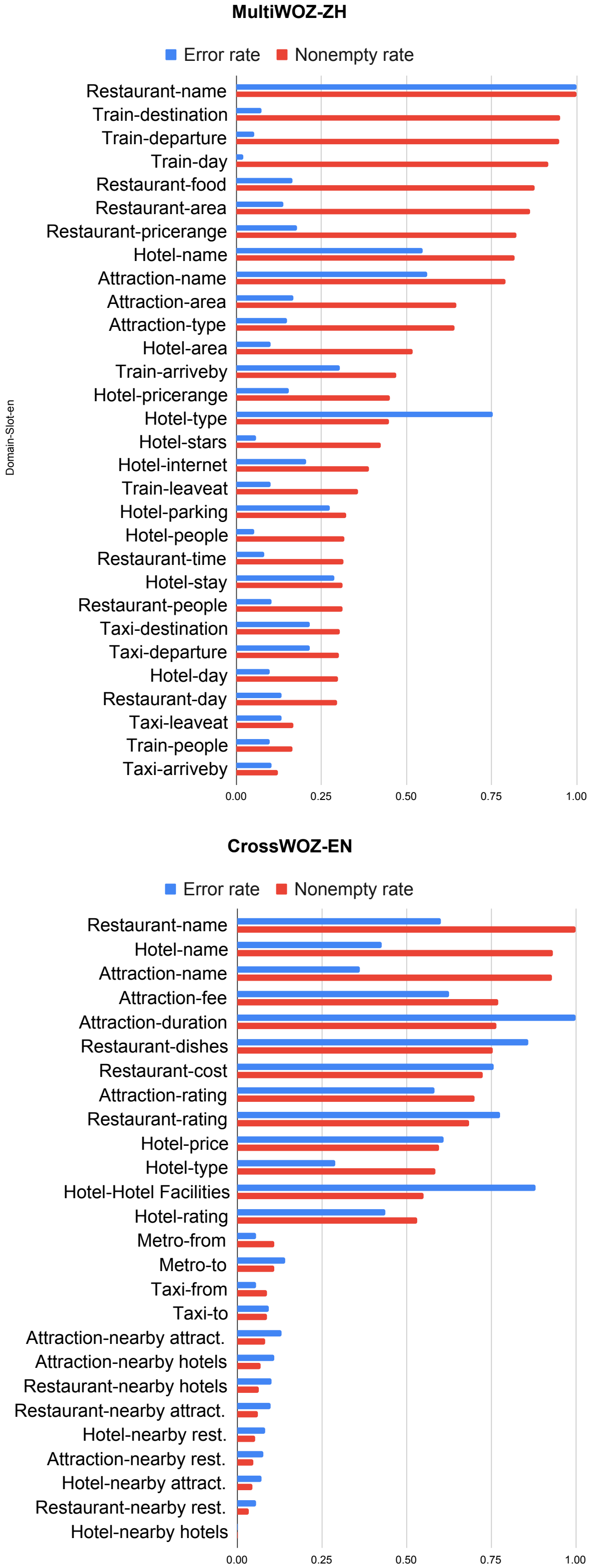


Figure 2: Error rates and nonempty rates (both are normalized by dividing the maximum) of all slots for MultiWOZ and CrossWOZ.

transformer-based models and have achieved significant performance improvements over the last year. In the cross-lingual dialog state tracking task, the participants built DST models with the training set in the rich-resource language and test set in the low-resource language. Interestingly, all the best submissions are trained on monolingual machine translated data instead of using both the original data and its translations, leaving the best approaches for model language portability as the future research topic.

References

Bao, S.; He, H.; Wang, F.; Wu, H.; Wang, H.; Wu, W.; Guo, Z.; Liu, Z.; and Xu, X. 2020. PLATO-2: Towards Building an Open-Domain Chatbot via Curriculum Learning. *arXiv preprint arXiv:2006.16779* .

Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; and Hu, G. 2019. Pre-Training with Whole Word Masking for Chinese BERT. *arXiv preprint arXiv:1906.08101* .

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Eric, M.; Goel, R.; Paul, S.; Sethi, A.; Agarwal, S.; Gao, S.; and Hakkani-Tur, D. 2019. MultiWOZ 2.1: Multi-Domain Dialogue State Corrections and State Tracking Baselines. *arXiv preprint arXiv:1907.01669* URL https://arxiv.org/abs/1907.01669.

Gao, J.; Peng, B.; Li, C.; Li, J.; Shayandeh, S.; Liden, L.; and Shum, H.-Y. 2020. Robust Conversational AI with Grounded Text Generation. *arXiv preprint arXiv:2009.03457* .

Gunasekara, C.; Kim, S.; D'Haro, L. F.; Rastogi, A.; Chen, Y.-N.; Eric, M.; Hedayatnia, B.; Gopalakrishnan, K.; Liu, Y.; Huang, C.-W.; Hakkani-Tür, D.; Li, J.; Zhu, Q.; Luo, L.; Liden, L.; Huang, K.; Shayandeh, S.; Liang, R.; Peng, B.; Zhang, Z.; Shukla, S.; Huang, M.; Gao, J.; Mehri, S.; Feng, Y.; Gordon, C.; Alavi, S. H.; Traum, D.; Eskenazi, M.; Beirami, A.; Eunjoon; Cho; Crook, P. A.; De, A.; Geramifard, A.; Kottur, S.; Moon, S.; Poddar, S.; and Subba, R. 2020. Overview of the Ninth Dialog System Technology Challenge: DSTC9.

Gururangan, S.; Marasovic, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 8342–8360. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.740/.

Ham, D.; Lee, J.; Jang, Y.; and Kim, K. 2020. End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 583–592. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.54/.

Heck, M.; van Niekerk, C.; Lubis, N.; Geishauser, C.; Lin, H.-C.; Moresi, M.; and Gavsi'c, M. 2020. TripPy: A Triple Copy Strategy for Value Independent Neural Dialog State Tracking. In *SIGdial*.

Hosseini-Asl, E.; McCann, B.; Wu, C.-S.; Yavuz, S.; and Socher, R. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796* .

Huang, H.; Liang, Y.; Duan, N.; Gong, M.; Shou, L.; Jiang, D.; and Zhou, M. 2019. Unicoder: A Universal Language Encoder by Pre-training with Multiple Cross-lingual Tasks. In *EMNLP/IJCNLP*.

Kim, S.; D'Haro, L. F.; Banchs, R. E.; Williams, J. D.; Henderson, M.; and Yoshino, K. 2016. The fifth dialog state tracking challenge. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, 511–517. IEEE.

Kim, S.; Galley, M.; Gunasekara, R. C.; Lee, S.; Atkinson, A.; Peng, B.; Schulz, H.; Gao, J.; Li, J.; Adada, M.; Huang, M.; Lastras, L. A.; Kummerfeld, J. K.; Lasecki, W. S.; Hori, C.; Cherian, A.; Marks, T. K.; Rastogi, A.; Zang, X.; Sunkara, S.; and Gupta, R. 2019. The Eighth Dialog System Technology Challenge. *CoRR* abs/1911.06394. URL http://arxiv.org/abs/1911.06394.

Kim, S.; Yang, S.; Kim, G.; and Lee, S.-W. 2020. Efficient Dialogue State Tracking by Selectively Overwriting Memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 567–582. Online: Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.53. URL https://www.aclweb.org/anthology/2020.acl-main.53.

Lample, G.; and Conneau, A. 2019. Cross-lingual Language Model Pretraining. In *NeurIPs*. URL http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf.

Lee, H.; Lee, J.; and Kim, T.-Y. 2019. SUMBT: Slot-Utterance Matching for Universal and Scalable Belief Tracking. In *Proc. Conf. Association for Computational Linguistics (ACL)*.

Lee, S.; Zhu, Q.; Takanobu, R.; Zhang, Z.; Zhang, Y.; Li, X.; Li, J.; Peng, B.; Li, X.; Huang, M.; and Gao, J. 2019. ConvLab: Multi-Domain End-to-End Dialog System Platform. In Costa-jussà, M. R.; and Alfonseca, E., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, 64–69. Association for Computational Linguistics. doi:10.18653/v1/p19-3011. URL https://doi.org/10.18653/v1/p19-3011.

Lei, W.; Jin, X.; Kan, M.-Y.; Ren, Z.; He, X.; and Yin, D. 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)*, 1437–1447. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/P18-1133. URL https://www.aclweb.org/anthology/P18-1133.

Li, J.; Peng, B.; Lee, S.; Gao, J.; Takanobu, R.; Zhu, Q.; Huang, M.; Schulz, H.; Atkinson, A.; and Adada, M. 2020. Results of the multi-domain task-completion dialog challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*.

Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; and Zettlemoyer, L. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *ArXiv* abs/2001.08210.

Mrksic, N.; Vulic, I.; Séaghdha, D. Ó.; Leviant, I.; Reichart, R.; Gasic, M.; Korhonen, A.; and Young, S. 2017. Semantic Specialization of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints. *Transactions of the Association for Computational Linguistics* 5: 309–324.

Peng, B.; Li, C.; Li, J.; Shayandeh, S.; Liden, L.; and Gao, J. 2020. SOLOIST: Few-shot Task-Oriented Dialog with A Single Pre-trained Auto-regressive Model. *arXiv preprint arXiv:2005.05298* .

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2018. Language Models are Unsupervised Multitask Learners. http://bit.ly/gpt_openai.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8): 9.

Schatzmann, J.; Thomson, B.; Weilhammer, K.; Ye, H.; and Young, S. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, 149–152.

Schuster, S.; Gupta, S.; Shah, R.; and Lewis, M. 2019. Cross-lingual Transfer Learning for Multilingual Task Oriented Dialog. In *NAACL-HLT*.

Shan, Y.; Li, Z.; Zhang, J.; Meng, F.; Feng, Y.; Niu, C.; and Zhou, J. 2020. A Contextual Hierarchical Attention Network with Adaptive Objective for Dialogue State Tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6322–6333. Online: Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.563. URL https://www.aclweb.org/anthology/2020.acl-main.563.

Takanobu, R.; Zhu, Q.; Li, J.; Peng, B.; Gao, J.; and Huang, M. 2020. Is Your Goal-Oriented Dialog Model Performing Really Well? Empirical Analysis of System-wise Evaluation. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 297–310. 1st virtual meeting: Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.sigdial-1.37.

Wang, K.; Tian, J.; Wang, R.; Quan, X.; and Yu, J. 2020a. Multi-Domain Dialogue Acts and Response Co-Generation. *arXiv preprint arXiv:2004.12363* .

Wang, Y.; Ke, P.; Zheng, Y.; Huang, K.; Jiang, Y.; Zhu, X.; and Huang, M. 2020b. A large-scale chinese short-text conversation dataset. In *CCF International Conference on Natural Language Processing and Chinese Computing*, 91–103. Springer.

Williams, J. D.; Atui, K. A.; and Zweig, G. 2017. Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 665–677.

Zhu, Q.; Huang, K.; Zhang, Z.; Zhu, X.; and Huang, M. 2020a. CrossWOZ: A Large-Scale Chinese Cross-Domain Task-Oriented Dialogue Dataset. *Transactions of the Association for Computational Linguistics* URL https://arxiv.org/abs/2002.11893.

Zhu, Q.; Zhang, Z.; Fang, Y.; Li, X.; Takanobu, R.; Li, J.; Peng, B.; Gao, J.; Zhu, X.; and Huang, M. 2020b. ConvLab-2: An Open-Source Toolkit for Building, Evaluating, and Diagnosing Dialogue Systems. In Çelikyilmaz, A.; and Wen, T., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, 142–149. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-demos.19/.

Example Dialogs

Table 8 and 9 include the example test dialogs of MultiWOZ and CrossWOZ respectively for the cross-lingual dialog state tracking task.

Table 8: An example dialog from the test set for MultiWOZ (en→zh) sub-task.

Speaker	Utterance	Dialog State Update

User	Hello! I am looking for a local guesthouse in the centre.
	你好！我在市中心找一家本地宾馆。
		hotel: {area: centre, type: guesthouse}
		旅馆: {区域: 中心, 类型: 宾馆}

System	OK. I am glad to recommend Alexander Bed and Breakfast to you. 好的。这边很高兴向您推荐亚历山大住宿加早餐旅馆。	
User	Where is it? 它在哪里?	hotel: {name: alexander bed and breakfast} 旅馆: {名称: 亚历山大住宿加早餐旅馆}
System	It is at 56 Saint Barnabas Road. 圣巴纳巴斯路56号。	

User	I also wish to have a meal in a local European restaurant in the centre.
	我还想在市中心的一家本地欧洲餐厅吃饭。
		restaurant: {food: european, area: centre}
		餐厅: {食物: 欧洲的, 区域: 中心}

System	You can choose Eraina. 您可以选择伊莱娜。	

User	Please give me its address.
	那请给我它的地址。
		restaurant: {name: eraina}
		餐厅: {名称: 伊莱娜}

System	It is in Free School Lane City Centre. 市中心自由校园巷。	
User	Ok. I'll go there. I need to book a taxi from Alexander Bed and Breakfast to Eraina after 07:00. 好。那我去那里。我还要预订一辆7:00时从亚历山大住宿加早餐旅馆到伊莱娜的出租车。	taxi: {leaveAt: 07:00, destination: eraina, departure: alexander bed and breakfast} 出租车: {出发时间: 07:00, 目的地: 伊莱娜, 出发地: 亚历山大住宿加早餐旅馆}
System	Well. I find a yellow Skoda. 好的。是一辆黄色的斯柯达。	
User	How about its phone number? 它的电话号码是多少?	No update
System	It is 78519675253. 78519675253。	
User	Thank you for your help. Bye! 谢谢你帮忙。再见！	No update
System	A pleasure. Bye bye! 我很乐意。再见！	

Table 9: An example partial dialog from the test set for CrossWOZ (zh→en) sub-task.

Speaker	Utterance	Dialog State Update

User	你好啊，我想请我朋友吃香椿拌豆腐，哈哈，你能帮我找一家评分是5分的餐馆吗?
	Hello, I’d like to treat my friend to Tofu with Chinese toon. Haha, can you find a restaurant with a rating of 5 points?
		餐馆: {推荐菜: 香椿拌豆腐, 评分: 5分}
		restaurant: {dishes: Tofu with Chinese toon, rating: 5 points}

System	我只查到了太后村98号院，四渡餐厅可以做这个菜，不过这两家餐馆都没有评分呀。 I only found that Taihou Village Yard 98 and Sidu Restaurant providing this dish. But the two don’t show their rating.	

User	这样啊，那我们去四渡餐厅尝尝吧，你知道他家的地址是在哪里吗?
	Well, we will go to Sidu Restaurant. Do you know the address of it?
		餐馆: {名称: 四渡餐厅, 评分: “”}
		restaurant: {name: Sidu Restaurant, rating: “”}

System	他家登记的地址是涞宝路，你可以去找一下。 Its registered address is Laibao Road. You can find it.	
User	好吧，麻烦你把他家电话发给我，找不到我可以打电话问。 Well, please send me its telephone number. If I can’t find, I can call them to get help.	No update
System	嗯，他们的电话是61344195，你可以咨询这部电话。 Well, their telephone number is 61344195. You can call them to ask.	

User	好的，周末我要跟同学去小玩一番，你能帮我们推荐一个门票价格是20-50元的好玩的景点吗?
	OK, I’m going to have fun with my classmate on the weekend. Can you recommend an attraction with a per-person cost of 20-50 yuan?
		景点: {门票: 20-50元}
		attraction: {fee: 20-50 yuan}

System	那就去趟鸟巢（国家体育场）吧，这个地方的热度很高。 You can go to Bird’s Nest (National Stadium), which is very popular.	

User	好呀，正好我们还没去过呢，你看下他旁边有什么别的景点没有?
	Well, we haven’t been there yet. Can you find some other attractions next to it?
		景点: {名称: 鸟巢（国家体育场）}
		attraction: {name: Bird’s Nest (National stadium)}

System	他旁边景点很多，像国家奥林匹克体育中心, 中国科学技术馆, 水立方嬉水乐园都是很有人气的景点哟。 There are many attractions near it, such as National Olympic Sports Center, China Science and Technology Museum and Water Cube Water Park. They are very hot.	
User	行，时间够用我们在去别处走走，对了，我还想问下这个景点周边有酒店吗? 我想帮我同学安排下住宿。 OK, if allows we will visit other places. By the way, is there a hotel around it? I want to arrange a hotel for my classmate.	No update
System	他旁边有北京国家会议中心大酒店, 锦江之星(北京奥体中心店), 北京亚奥国际酒店等几家酒店，你们可以去看看。 Many hotels are near it including China National Convention Center, Beijing Olympic Sports Center and Beijing Ya’ao International Hotel Beijing. You can pick one and have a look.	

User	这几家好像都不行呀，你重新帮我找一家价格是500-600元，评分是4.5分以上的酒店吧。
	It seems that none of them meeting my needs. Please find me a hotel with a price of 500-600 yuan and a rating of 4.5 points or above.
		酒店: {价格: 500-600元, 评分: 4.5分以上}
		hotel: {price: 500-600 yuan, rating: 4.5 points or above}

System	我推荐一家酒店吧，北京紫玉饭店就可以的。 Let me recommend you one. Ziyu Hotel Beijing is OK.	