Learning Low-Resource End-To-End Goal-Oriented Dialog for Fast and Reliable System Deployment

Yinpei Dai[†], **Hangyu Li**[†], **Chengguang Tang**[†], **Yongbin Li**[†]*, **Jian Sun**[†], **Xiaodan Zhu**[‡]
[†]Alibaba Group, Beijing
[‡]Ingenuity Labs Research Institute & ECE, Queen's University
{yinpei.dyp,hangyu.lhy,chengguang.tcg}@alibaba-inc.com
{shuide.lyb,jian.sun}@alibaba-inc.com, zhu2048@gmail.com

Abstract

Existing end-to-end dialog systems perform less effectively when data is scarce. To obtain an acceptable success in real-life online services with only a handful of training examples, both fast adaptability and reliable performance are highly desirable for dialog systems. In this paper, we propose the Meta-Dialog System (MDS), which combines the advantages of both meta-learning approaches and human-machine collaboration. We evaluate our methods on a new extended-bAbI dataset and a transformed MultiWOZ dataset for low-resource goal-oriented dialog learning. Experimental results show that MDS significantly outperforms non-meta-learning baselines and can achieve more than 90% per-turn accuracies with only 10 dialogs on the extended-bAbI dataset.

1 Introduction

End-to-end neural models have shown a great potential in building flexible goal-oriented dialog systems. They can be directly trained on past dialogs without any domain-specific handcrafting, which makes it easy to automatically scale up to new domains (Bordes et al., 2017). However, these models are normally data-hungry and have only been successfully applied to domains with rich datasets (Perez et al., 2017; Luo et al., 2019; Kim et al., 2019).

In real-world scenarios, common issues with end-to-end dialog models include: (1) the shortage of proper training dialogs because of the high cost of data collection and cleaning, i.e., the *data scarcity* problem (Zhao and Eskenazi, 2018), and (2) a large gap between limited data and unknown online test examples, i.e., the *covariate shift* effect (Liu et al.). Such problems can lead to a significant performance degradation in dialog systems, which may harm the users' experience and result in loss of customers in commercial applications. Therefore, both *fast adaptability* and *reliable performance* are strongly desirable for practical system deployment. Fast adaptability reflects the efficiency of adapting dialog systems to domains with low-resource data. Reliable performance reflects the robustness of handling unpredictable user behaviors in online services.

To boost the online performance of dialog systems, there have been some recent work (Rajendran et al., 2019; Wang et al., 2019; Lu et al., 2019) on designing end-to-end models in a human-machine joint-teaming manner. For instance, the dialog system in (Rajendran et al., 2019) can identify an ongoing dialog during testing when the system might fail and transfer it to a human agent. But all these methods are trained with sufficient data, which hinders the possibility of rapidly prototyping the models in new domains with restricted resources.

In this paper, we formulate the low-resource goal-oriented dialog learning as a few-shot learning problem, where a limited numbers of dialogs are used for training and the remaining for the test. We propose the Meta-Dialog System (MDS), an end-to-end human-machine teaming framework optimized by the model-agnostic meta-learning (MAML) algorithm (Finn et al., 2017). In general, MDS learns to make prediction and requests human by finding good initial parameters, which can be adapted to new tasks fast and reliably by using fewer dialogs. We evaluate our methods on a new multi-domain dialog dataset called *extended-bAbI*. Results show that MDS achieves obvious performance improvement over baselines and attains more than 90% per-turn accuracy on new domains with only 10 dialogs. We also perform experiments on MultiWOZ dataset (Eric et al., 2019) which has been transformed into simplified bAbI format and observe similar superior results with MDS.

*Corresponding author

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 609–618
July 5 - 10, 2020.

In summary, the main contributions of this paper are three-fold: (1) To the best of our knowledge, this is the first study on applying meta-learning to retrieval-based end-to-end goal-oriented dialog systems; (2) we leverage the MAML algorithm to optimize a human-machine collaborative dialog system and show very promising results on the low-resource dialog tasks; and (3) we propose a new dataset and hope that can help bring forward the research in this area.

2 The Proposed Method

In this section, we first introduce the problem definition and our new dataset; we then elaborate the framework of MDS and meta-learning procedures.

Problem Definition. We focus on the retrieval-based goal-oriented dialog tasks (Perez et al., 2017), where a training data d_i usually contains a triple $(H_i, y_i, \mathcal{R})$. H_i denotes the dialog history consisting of all user utterances and system responses up to the current turn, $\mathcal{R}$ is a set of given candidate responses and y_i is the index of the correct response in $\mathcal{R}$. The main task is to train an end-to-end dialog model to predict y_i from $\mathcal{R}$ based on H_i.

Extended-bAbI Dataset. The original bAbI dataset (Bordes et al., 2017) is not suitable for low-resource settings due to the lack of domains and tasks. We extend it into a multi-domain dataset through complicated simulation rules and construct templates with a more diversity to raise the difficulty. There are 7 domains in total: *restaurant, flights, hotels, movies, music, tourism* and *weather*, each of which has its own ontology and the candidate response set. Similar to (Bordes et al., 2017), a complete dialog in extended-bAbI contains four phases of interactions: (1) the system asks for required attributes to constrain the search and issues the first API call; (2) the user updates their requests for revised API calls; (3) the system confirms for multiple times to determine the entity the user wants; (4) the user requests more attributes for extra information based on the final entity. The total number of dialogs is 21,000 and the detailed examples and statistics are given in Appendix A.1.

2.1 Model Architecture

In MDS, there is an *encoding module* to extract neural features of dialogs and a *policy module* to make system actions of either predicting responses or requesting human. All modules are jointly optimized with the MAML algorithm. The main framework of training MDS is shown in Figure 1.

Encoding Module. It contains a history encoder to compute the dialog state vector s_i for H_i and a response encoder to compute the response embedding r_j for the j-th response in $\mathcal{R}$. The dimensions of s_i and r_j are set as the same. In this paper, we use the MemN2N (Sukhbaatar et al., 2015) as the history encoder and a simple additive model for the response encoder, but many other models optimized by gradient descent may be applied here.

Policy Module. This module consists of a switch S that makes a binary decision whether to request human to select the response, and a response predictor P that predicts the right response itself if human is not requested. We assume that the response chosen by human is always correct.

For the optimization of P, the widely used large-margin cosine loss (Wang et al., 2018; Lin and Xu, 2019) is employed since it maximizes the decision margin in the angular space and is able to force the model to learn more discriminative deep features. Suppose a batch of training data is $\mathcal{D} = \{d_1, ...d_i, ..., d_{|\mathcal{D}|}\}$, then the formulation is:

$$\mathcal{L}_{\text{LMC}} = \sum_{i=1}^{|\mathcal{D}|} -\log \frac{e^{a\cdot(\cos(s_i, r_{y_i})-b)}}{e^{a\cdot(\cos(s_i, r_{y_i})-b)} + \sum_{j\neq y_i} e^{a\cdot\cos(s_i, r_j)}} \quad (1)$$

where $\cos(\cdot,\cdot)$ is a function that calculates the cosine similarity of two input vectors. a is the scaling factor and b is the cosine margin ($a = 30, m = 0.1$ in our experiments). In the test phase, the model predicts an answer according to the maximal cosine angle $y_i^* = \text{argmax}_j \cos(s_i, r_j)$.

The switch S is a neural binary classifier that also takes s_i and each r_j as input and calculate the decision probability of requesting human as follows:

$$w_{ij} = e^{s_i^\top W r_j} / \sum_{k=1}^{|\mathcal{R}|} e^{s_i^\top W r_k} \quad (2)$$

$$c_i = \sum_{j=1}^{|\mathcal{R}|} w_{ij} r_j \quad (3)$$

$$f_i = s_i \oplus c_i \quad (4)$$

$$p_i = \sigma(\text{FC}(f_i)) \quad (5)$$

where σ is the sigmoid function and $\oplus$ the concatenation function for vectors. FC$(\cdot)$ is a fully-connected neural network with one hidden layer that has half size of the input layer and is activated

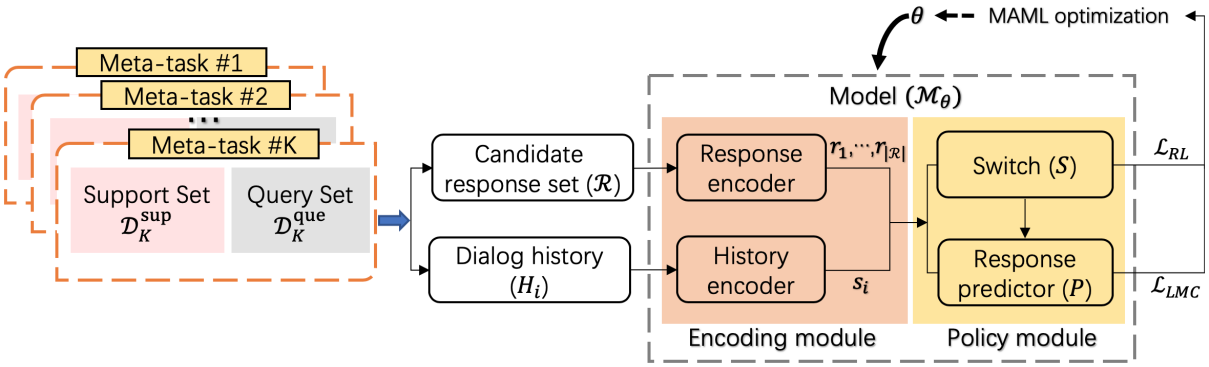

Figure 1: An overview of training the Meta-Dialog System.

by *tanh* function. $|\mathcal{R}|$ is the size of $\mathcal{R}$ and W is a trainable square matrix.

Learning to switch. Since there are no actual labels for S to indicate whether it is correct to ask human or not, some previous work (Woodward and Finn, 2016; Rajendran et al., 2019) proposes to use the REINFORCE algorithm (Williams, 1992) for weakly-supervised training, but their reward settings fail to penalize the case when the model asks human while it can give right prediction, which may lead to redundant requests. To consider this effect, we propose a new reward definition here. For the batch data $\mathcal{D}$, we calculate the F1 scores[1] for positive data and negative data, respectively, and take the average of them to get a scalar value $score(\mathcal{D})$. Then each data $d_i \in \mathcal{D}$ is assigned with a reward by computing an incremental value as below:

$$r_t = score(\mathcal{D}) - score(\mathcal{D} - d_i) \quad (6)$$

Through maximizing such rewards, the switch S learns to be more effective and asks human when it is necessary. The reinforcement learning loss for S is $\mathcal{L}_{\text{RL}} = \sum_{i=1}^{|\mathcal{D}|} -r_i \log p_i$, and the final loss of our model is $\mathcal{L} = \mathcal{L}_{\text{LMC}} + \mathcal{L}_{\text{RL}}$.

2.2 Training Procedure

We rewrite the final loss $\mathcal{L}$ as $\mathcal{L}(\mathcal{M}_\theta, \mathcal{D})$ for clarity, where $\mathcal{M}_\theta$ denotes the dialog model with trainable parameters θ and $\mathcal{D}$ is the batch data for training.

During meta-learning, we first choose one domain as the target domain and the rest as source domains. Then we uniformly sample K different domains $\mathcal{T} = \{\tau_1, \ldots, \tau_K\}$ from source domains as meta-tasks. For each meta-task τ_k, we sample N data as the support set $\mathcal{D}_k^{\text{sup}}$ and other N data with the same answers as the query set $\mathcal{D}_k^{\text{que}}$.

[1]Detailed explanations can be found in Appendix A.2.

Algorithm 1 Meta-learning for MDS

Input: The learning rates α, β
Output: optimal meta-learned model

```
Initialize model parameters θ randomly
while not converged do
    Sample T from source domains and prepare D_k^sup, D_k^que
    for each τ_k do
        Evaluate L(M_θ, D_k^sup)
        Compute θ'_k = θ − α∇_θ L(M_θ, D_k^sup)
        Evaluate L(M_θ'_k, D_k^que)
    end for
    Update θ ← θ − β∇_θ Σ_{k=1}^K L(M_θ'_k, D_k^que)
end while
```

$\mathcal{M}_\theta$ is first updated on support sets for each τ_k:

$$\theta'_k = \theta - \alpha \nabla_\theta \mathcal{L}(\mathcal{M}_\theta, \mathcal{D}_k^{\text{sup}}) \quad (7)$$

Then $\mathcal{M}_\theta$ is evaluated on each $\mathcal{D}_k^{\text{que}}$ with θ'_k respectively and is optimized as follows:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{k=1}^{K} \mathcal{L}(\mathcal{M}_{\theta'_k}, \mathcal{D}_k^{\text{que}}) \quad (8)$$

where α, β are learning rates. By training on multiple tasks via MAML, $\mathcal{M}_\theta$ can learn good initial parameters that is applicable on new tasks or domains (Finn et al., 2017; Mi et al., 2019). The algorithm is summarised in Algorithm 1.

After this meta-learning as pre-training, we fine-tune $\mathcal{M}_\theta$ on the target domain with the first L dialogs of its training set, where L is a small number. To mimic the situation of online testing, we evaluate $\mathcal{M}_\theta$ on the whole test sets and regard those unseen user utterances as new user behaviours.

3 Experiments and Results

In our experiments, we first verify the capability of MDS on our newly simulated dialog dataset

extended-bAbI, and then conduct extra evaluation on the more realistic dataset MultiWOZ 2.1 (Eric et al., 2019).

3.1 Setup

We select each domain as the target domain in turn and take the average of the results in all domains.

Metric. Following (Wang et al., 2019), we report the user-perceived per-turn accuracy ('per-turn accuracy' is used in the remainder of the paper), where the prediction of one turn is considered correct if the model either selects the right response by itself or asks human. To be fair, we also report the human request rate. The less the request rate and higher per-turn accuracy are, the more reliable the model performs online.

Implementation details. For the meta-learning, we use SGD for the inner loop and Adam for the outer loop with learning rate α=0.01 and β=0.001. The meta-task size K is 4 and the support or query set size N is 16. For the standard MLE training, we use Adam with a learning rate of 0.001 and set the batch size as 32. Both schemes are trained for a maximum of 5000 iterations with early stopping on the validation set. During fine-tuning on new domains, we use SGD with the learning rate 0.01 for all models and report the final results after fine-tuning 10 iterations on L training dialogs of the target domain, where L=0, 1, 5, 10. The word vector size is 25 and all MemN2Ns take 3 hops.

3.2 Baselines

We compare MDS with the following baselines:

- **Mem**: A MemN2N (Sukhbaatar et al., 2015) model trained with standard MLE.
- **MetaMem**: A MemN2N trained with MAML. Both Mem and MetaMem can not request human.
- **Mem+C**: A MemN2N model combined with a binary classifier in (Rajendran et al., 2019), which has different objective functions and optimization.
- **IDS**: The incremental dialog system used in (Wang et al., 2019), which requests human by estimating the uncertainty through a variational autoencoder.
- **MDS$_{\text{-switch}}$**: A MDS without the switch S.
- **MDS$_{\text{rand}}$**: A MDS whose switch is replaced with a random classifier that has the same request rate.
- **MDS$_{\text{mle}}$**: A MDS whose meta-learning optimization is replaced with standard MLE.

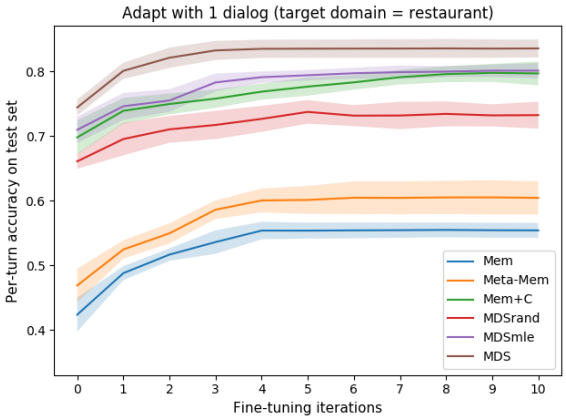

Figure 2: The per-turn accuracy of different methods on the test set during fine-tuning with 1 dialog adaptation where the target domain is *restaurant*.

3.3 Results on Extended-bAbI

Table 1 shows few-shot adaptation results for different methods. MDS significantly outperforms other models under all adaptation sizes of new dialogs and can achieve a 91.31% per-turn accuracy on average with only 10 new dialogs.

There is a gap between methods without the switch (such as Mem, MetaMem and MDS$_{\text{-switch}}$) and methods with the switch in Table 1, indicating that the switch S is crucial for improving the overall per-turn accuracy because of the human agent. However, without proper objective functions and meta-learning optimization, Mem+C and IDS[2] have poorer performances in both metrics than MDS even if they contain the switch module.

In the ablation study, we see a steady increase of about 10% per-turn accuracy from the comparison between MDS and MDS$_{\text{rand}}$, suggesting that the switch does identify intractable dialogs. MDS$_{\text{mle}}$ is the closest baseline to MDS, but we still observe an obvious improvement, which means joint optimization of S and P via meta-learning allows faster and better adaptation while maintaining similar request rates. Appendix A.3 illustrates detailed case studies for different methods.

To further investigate the adaptation process, we present the fine-tuning curves for different methods with 1 dialog adaptation in Figure 2. As it can be seen, MDS achieves the best accuracy at the beginning and converges fastest as well, showing that it can transfer on new tasks quickly by finding better parameter initialization.

[2]We only report the result of IDS with 10 dialog adaptation since its request rates are too high to be fair in other settings.

Method	No adaptation		Adapt with 1 dialog		Adapt with 5 dialogs		Adapt with 10 dialogs	
	accuracy	request	accuracy	request	accuracy	request	accuracy	request
Mem	32.28±1.86	n.a.	45.02±1.39	n.a.	64.07±0.76	n.a.	71.56±0.48	n.a.
MetaMem	39.45±1.13	n.a.	48.95±1.18	n.a.	65.57±0.69	n.a.	72.19±0.81	n.a.
Mem+C	58.74±2.89	37.34±5.23	68.27±2.19	34.83±4.35	81.41±2.26	36.96±5.05	87.46±2.07	38.09±5.29
IDS	-	-	-	-	-	-	90.91±4.29	83.98±6.43
$MDS_{\text{-switch}}$	41.03±0.98	n.a.	50.31±1.16	n.a.	65.72±1.13	n.a.	72.35±0.90	n.a.
MDS_{rand}	61.05±1.20	34.75	66.02±0.91	32.31	77.27±0.76	34.31	79.70±0.98	35.26
MDS_{mle}	59.89±3.11	**34.36±6.09**	69.40±2.25	32.46±4.06	83.04±2.07	**33.90±5.22**	88.13±1.63	35.28±5.08
MDS	**64.93±2.39**	34.75±5.87	**74.71±2.15**	**32.31±4.34**	**86.49±2.01**	34.31±4.36	**91.31±1.16**	**35.26±4.23**

Table 1: Few-shot results on the extended-bAbI dataset. The numbers represent the average of means and standard deviations of Task 5 in all target domains. Each experiment run 10 times with different seeds; 'n.a.' means no switch in the model; 'accuracy' is the user-perceived per-turn accuracy and 'request' is the request rate.

3.4 Results on MultiWOZ

MultiWOZ (Budzianowski et al., 2018) is a widely-used multi-domain Wizard-of-Oz dialog dataset spanning 7 distinct domains and containing 10k dialogs. This realistic dataset has been a standard benchmark for various dialog tasks such as belief tracking and policy optimization.

In our experiment, we use the corrected version MultiWOZ 2.1 (Eric et al., 2019) for evaluation. To translate the MultiWOZ dialogs into bAbI-format data, we first delexicalize the slot-values in user utterances using dialog labels, and then produce a set of canonical system acts as the candidate responses by simplifying the original dialog acts. Only dialogs containing single domain are used in our experiments and a MultiWOZ dialog sample is given in Appendix A.4.

Table 2 shows the adaptation results for different models on MultiWOZ 2.1. It can be seen that MDS still largely outperforms other models with the adaptation of 10 dialogs. The degradation of per-turn accuracy from extended-bAbI to MultiWOZ is reasonable since the user utterance is more diverse and the dialog policy is more flexible.

Method	Adapt with 10 dialogs	
	accuracy	request
Mem	56.87±1.63	n.a.
MetaMem	62.78±2.05	n.a.
Mem+C	80.59±3.13	38.18±5.01
$MDS_{\text{-switch}}$	64.50±3.75	n.a.
MDS_{rand}	74.78±4.35	38.34
MDS_{mle}	80.92±3.02	**37.91±4.20**
MDS	**83.52±3.30**	38.34±6.96

Table 2: Few-shot test results on MultiWOZ 2.1.

4 Related Work

End-to-end neural approaches of building dialog systems have attracted increasing research interest. The work of (Bordes et al., 2017) is the first attempt to solve goal-oriented dialog tasks with end-to-end models. Further improvements has been made in (Williams et al., 2017) to combine explicit domain-specific knowledge and implicit RNN features. Luo et al. (2019) take user personalities into consideration for better user satisfaction. Rajendran et al. (2018) learn dialogs with multiple possible answers. Our work is inspired by the work of (Rajendran et al., 2019; Wang et al., 2019), which propose to solve unseen user behaviors through human-machine teamwork. The research of (Liu et al.; Chen et al., 2017; Lu et al., 2019) also show the advantages of incorporating the role of human to teach online. However, dialog learning in low-resource scenarios has not been investigated.

Meta-learning aims to learn new tasks rapidly with a few training examples (Sung et al., 2018; Finn et al., 2017), which fits well to our task. There have been some work applying meta-learning to other tasks in dialog research, such as that in (Dou et al., 2019; Geng et al., 2019) for natural language understanding and (Qian and Yu, 2019; Mi et al., 2019) for natural language generation.

5 Conclusion and Future Work

In this paper, we leverage the MAML algorithm to optimize a human-machine collaborative dialog system, which shows good results for both fast adaptability and reliable performance. In the future, we plan to use more powerful encoders and evaluate our methods on real dialog data.

Acknowledgments

The research of the last author is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. *ICLR*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, 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*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Lu Chen, Xiang Zhou, Cheng Chang, Runzhe Yang, and Kai Yu. 2017. Agent-aware dropout DQN for safe and efficient on-line dialogue policy learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2454–2464, Copenhagen, Denmark. Association for Computational Linguistics.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1192–1197, Hong Kong, China. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. *arXiv preprint*, 1907.01669.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3902–3911, Hong Kong, China. Association for Computational Linguistics.

Byoungjae Kim, KyungTae Chung, Jeongpil Lee, Jungyun Seo, and Myoung-Wan Koo. 2019. A bi-lstm memory network for end-to-end goal-oriented dialog learning. *Computer Speech & Language*, 53:217–230.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5496, Florence, Italy. Association for Computational Linguistics.

Bing Liu, Gokhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry Heck. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Yichao Lu, Manisha Srivastava, Jared Kramer, Heba Elfardy, Andrea Kahn, Song Wang, and Vikas Bhardwaj. 2019. Goal-oriented end-to-end conversational models with profile features in a real-world setting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 48–55, Minneapolis - Minnesota. Association for Computational Linguistics.

Liangchen Luo, Wenhao Huang, Qi Zeng, Zaiqing Nie, and Xu Sun. 2019. Learning personalized end-to-end goal-oriented dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6794–6801.

Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. 2019. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. *AAAI*.

Julien Perez, Y-Lan Boureau, and Antoine Bordes. 2017. Dialog system & technology challenge 6 overview of track 1-end-to-end goal-oriented dialog learning. *Dialog System Technology Challenges*, 6.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2639–2649, Florence, Italy. Association for Computational Linguistics.

Janarthanan Rajendran, Jatin Ganhotra, and Lazaros C Polymenakos. 2019. Learning end-to-end goal-oriented dialog with maximal user task success and minimal human agent use. *Transactions of the Association for Computational Linguistics*, 7:375–386.

Janarthanan Rajendran, Jatin Ganhotra, Satinder Singh, and Lazaros Polymenakos. 2018. Learning end-to-end goal-oriented dialog with multiple answers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3834–3843, Brussels, Belgium. Association for Computational Linguistics.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition, pages 1199–1208.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274.

Weikang Wang, Jiajun Zhang, Qian Li, Mei-Yuh Hwang, Chengqing Zong, and Zhifei Li. 2019. Incremental learning from scratch for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3710–3720, Florence, Italy. Association for Computational Linguistics.

Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. 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)*, pages 665–677, Vancouver, Canada. Association for Computational Linguistics.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Mark Woodward and Chelsea Finn. 2016. Active one-shot learning. *NIPS (Deep Reinforcement Learning Workshop)*.

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

A Appendices

A.1 Extended-bAbI dialog dataset

We extend bAbI dataset (Bordes et al., 2017) into a larger dialog dataset consisting of multiple domains, where each domain has its own ontology and the candidate response set. The main task is a reponse retrieval problem, where the dialog system needs to select the right response for current dialog history from the given candidate response set. The size of candidate sets in each domain are shown in Table 3. The total number of dialogs for each task is 3000 (1500/500/1000 for train/dev/test set respectively). More statistics are given in Table 4. Detailed dialog samples of extended-bAbI can be found in Table 6.

Domain	# responses
restaurant	333
flights	71
hotels	472
movies	68
music	56
tourism	47
weather	22

Table 3: The number of candidate responses in each domain.

Item	number
# of domains	7
# of dialog tasks	35
# of total system responses	1069
# of total templates for user utterances	685
# vocabulary size	386
# of sentences per dialog	12.4
# of words per sentence	4.9

Table 4: Statistics of extended-bAbI dataset.

A.2 Reward Settings

(Woodward and Finn, 2016; Rajendran et al., 2019) defined rewards for S as follows:

- R_{req}: if human is requested
- R_{cor}: if human is not requested and the model prediction is correct
- R_{inc}: if human is not requested and the model prediction is wrong

This kind of reward setting did not penalize the case that human is requested but model prediction is correct, and the value of rewards is fixed during reinforcement learning.

Here we propose better reward definition. Let a batch of data be $\mathcal{D} = \{d_1, d_2, ...d_i, ...\}$. The label of each data is set as positive if the response predictor P selects a wrong answer and negative if not. The prediction of each data is set as positive if the switch S chooses human and negative if not. Then we calculate the F1 scores for positive data and negative data separately and take the average of them to get a scalar value $score(D)$. Each data d_i in the batch $\mathcal{D}$ is assigned to a reward value by computing the difference of total scores with or without d_i

$$r_i = score(\mathcal{D}) - score(\mathcal{D} - d_i)$$

In this way, the reward function is less sensitive to the model prediction accuracy and can self-adjust dynamically during training. Both situations that human is asked while the model predicts correct and human is not asked while the model predicts wrong are given negative rewards.

A.3 Case Studies

There are two sample dialog generated by different models adapted with 10 dialogs on extended-bAbI in Table 5. It shows that MDS can predict the response correctly and choose to request human properly.

A.4 MultiWOZ Data

Table 7 shows a bAbI-format dialog example transferred from MultiWOZ. The data and code for dialog transformation of MultiWOZ and dialog generation of exteneded-bAbI are attached in the supplemental material.

sample 1 target domain = music		
Usr	how do you do	
Sys	hello what can i do for you	
Usr	i 'd like to hear taylor 's music	
Sys	what kind of music do you like	
Usr input	i would like to hear some country songs	
Correct answer	api_call taylor country	
Predictions:		
Mem	hello what can i do for you	[incorrect]
MetaMem	what kind of music do you like	[incorrect]
Mem+C	what do you think of this song:entityOrder=0	[incorrect][no-request]
MDS	api_call taylor country	[correct][request]
sample 2 target domain = weather		
Usr	hello	
Sys	hello how i can help you	
Usr	i want to know the weather in guangzhou	
Sys	which date do you want to check	
Usr input	i want to know the weather today	
Correct answer	api_call london today	
Predictions:		
Mem	which date do you want to check	[incorrect]
MetaMem	api_call guangzhou today	[correct]
Mem+C	hello how i can help you	[incorrect][no-request]
MDS	api_call guangzhou today	[correct][no request]

Table 5: Prediction of different methods on two sampled dialogs. The flag [incorrect] or [correct] indicate the result of response prediction. The flag [request] or [no-request] indicate whether human is requested or not.

Domain: music	
Usr:	what 's up
Sys:	how can i help you
Usr:	how can i find some music
Sys:	any singer in your mind
Usr:	i 'd like some songs by my idol taylor
Sys:	what kind of music do you like
Usr:	do you have country music
Sys:	api_call taylor country
Usr:	nice
Sys:	what do you think of this song : entityOrder=0
Usr:	do you have other suggestions
Sys:	how about another song : entityOrder=1
Usr:	let 's do it
Sys:	great let me play this song for you
Usr:	what is the album of the music
Sys:	here you are entityOrder=3 album
Usr:	thanks a lot
Sys:	can i help you with anything else
Usr:	no thank you
Sys:	you are welcome
Domain: weather	
Usr:	i want to know the weather tomorrow
Sys:	which city would you like to check
Usr:	hangzhou please
Sys:	api_call hangzhou tomorrow
Usr:	what 's the weather today
Sys:	api_call hangzhou today
Usr:	thanks
Sys:	it 's my pleasure

Table 6: Two dialog samples from *music* and *weather* domains on extended-bAbI.

Domain: restaurant	
Usr:	I would like a v_price restaurant which serves v_food food.
Sys:	api_call v_food v_price
Usr:	What is their address and what part of town are they in?
Sys:	Inform Addr Area
Usr:	Sorry what type of food do they serve?
Sys:	Inform Food
Usr:	Thank you. goodbye.
Sys:	general_bye

Table 7: An example of bAbI-format MultiWOZ dialog.