Meta-Learning for Low-resource Natural Language Generation in Task-oriented Dialogue Systems

Fei Mi1*, Minlie Huang2*, Jiyong Zhang3 and Boi Faltings1

1Artificial Intelligence Laboratory, École Polytechnique Fédérale de Lausanne (EPFL)
2Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University
3Depart of Automation, Hangzhou Dianzi University

fei.mi@epfl.ch, aihuang@tsinghua.edu.cn, jzhang@hdu.edu.cn, boi.faltings@epfl.ch

Abstract

Natural language generation (NLG) is an essential component of task-oriented dialogue systems. Despite the recent success of neural approaches for NLG, they are typically developed for particular domains with rich annotated training examples. In this paper, we study NLG in a low-resource setting to generate sentences in new scenarios with handful training examples. We formulate the problem from a meta-learning perspective, and propose a generalized optimization-based approach (Meta-NLG) based on the well-recognized model-agnostic meta-learning (MAML) algorithm. Meta-NLG defines a set of meta tasks, and directly incorporates the objective of adapting to new low-resource NLG tasks into the meta-learning optimization process. Extensive experiments are conducted on a large multi-domain dataset (MultiWOZ) with diverse linguistic variations. We show that Meta-NLG significantly outperforms other training procedures in various low-resource configurations. We analyze the results, and demonstrate that Meta-NLG adapts extremely fast and well to low-resource situations.

1 Introduction

As an essential part of a task-oriented dialogue system [Wen et al., 2016b], the task of natural language generation (NLG) is to produce a natural language utterance containing the desired information given a semantic representation consisting of dialogue act types with a set of slot-value pairs. Conventional methods using hand-crafted rules often generates monotonic utterances and it requires substantial amount of human engineering work. Recently, various neural approaches [Wen et al., 2015c; Tran and Nguyen, 2017; Tseng et al., 2018] have been proposed to generate accurate, natural and diverse utterances. However, these methods are typically developed for particular domains. Moreover, they are often data-intensive to train. The high annotation cost prevents developers to build their own NLG component from scratch. Therefore, it is extremely useful to train a NLG model that can be generalized to other NLG domains or tasks with a reasonable amount of annotated data. This is referred to low-resource NLG task in this paper.

Recently, some methods have been proposed for low-resource NLG tasks. Apart from the simple data augmentation trick [Wen et al., 2016a], specialized model architectures, including conditional variational auto-encoders (CVAEs, [Tseng et al., 2018; Tran and Nguyen, 2018a; Tran and Nguyen, 2018b]) and adversarial domain adaptation critics [Tran and Nguyen, 2018a], have been proposed to learn domain-invariant representations. Although promising results were reported, we found that datasets used by these methods are simple which tend to enumerate many slots and values in an utterance without much linguistic variations. As a consequence, over-fitting the slots and values in the low-resource target domain could even outperform those versions trained with rich source domain examples [Tran and Nguyen, 2018b]. Fortunately, there is a new large-scale dialog dataset (MultiWOZ, [Budzianowski et al., 2018]) that contains a great variety of domains and linguistic patterns that allows us to conduct extensive and meaningful experimental analysis for low-resource NLG tasks.

In this paper, instead of casting the problem as model-based approaches, we propose a generalized optimization-based meta-learning approach to directly enhance the optimization procedure for the low-resource NLG task. We start by arguing that a recently proposed model-agnostic meta-learning algorithm (MAML, [Finn et al., 2017]) is a nice fit to the low-resource NLG task. Then, we proposed a generalized NLG algorithm called Meta-NLG based on MAML by viewing languages in different domains or dialog act types as separate Meta NLG tasks. Following the essence of MAML, the goal of Meta-NLG is to learn a better initialization of model parameters that facilitates fast adaptation to new low-resource NLG scenarios. As Meta-NLG is model-agnostic as long as the model can be optimized by gradient descent, we could apply it to any existing NLG models to optimize them in a way that adapt better and faster to new low-resource tasks.

The main contribution of this paper is two-fold:

- We propose a meta-learning algorithm Meta-NLG based on MAML for low-resource NLG tasks. Since Meta-NLG is model-agnostic, it is applicable to many other NLG models. To the best of our knowledge, this is the
first study of applying meta-learning to NLG tasks.

- We extensively evaluate Meta-NLG on the largest multi-domain dataset (MultiWoZ) with various low-resource NLG scenarios. Results show that Meta-NLG significantly outperforms other optimization methods in various configurations. We further analyze the superior performance of Meta-NLG, and show that it indeed adapts much faster and better.

2 Background

2.1 Natural Language Generation (NLG)

Neural models have recently shown promising results in tackling NLG tasks for task-oriented dialog systems. Conditioned on some semantic representation called dialog act (DA), a NLG model decodes an utterance word by word, and the probability of generating an output sentence of length $T$ is factorized as follows:

$$f_\theta = P(Y|d; \theta) = \prod_{t=1}^{T} P(y_t|y_0, ..., y_{t-1}, d; \theta)$$  (1)

$f_\theta$ is the NLG model parameterized by $\theta$, and $d$ is the DA of sentence $Y = (y_0, y_1, ..., y_T)$. For example, $d$ is a one-hot representation of a DA “Inform(name=The Oak Bistro, food=British)”. “Inform” (DA type) controls the sentence functionality, and “name” and “food” are two involved slots. A realization utterance $Y$ could be “There is a restaurant called [The Oak Bistro] that serves [British] food.”. Each sentence might contain multiple DA types. A series of neural methods have been proposed, including HLSTM [Wen et al., 2015a], SCLSTM [Wen et al., 2015c], Enc-Dec [Wen et al., 2015b] and RALSTM [Tran and Nguyen, 2017].

2.2 Low-resource NLG

The goal of low-resource NLG is to fine-tune a pre-trained NLG model on new NLG tasks (e.g., new domains) with a small amount of training examples. [Wen et al., 2016a] proposed a “data counterfeiting” method to augment the low-resource training data in the new task without modifying the model or training procedure. [Tseng et al., 2018] proposed a semantically-conditioned variational autoencoder (SCVAE) learn domain-invariant representations feeding to SCLSTM. They shown that it improves SCLSTM in low-resource settings. [Tran and Nguyen, 2018b] adopted the same idea as in [Tseng et al., 2018]. They used two conditional variational autoencoders to encode the sentence and the DA into two separate latent vectors, which are fed together to the decoder RALSTM [Tran and Nguyen, 2017]. They later designed two domain adaptation critics with an adversarial training algorithm [Tran and Nguyen, 2018a] to learn an indistinguishable latent representation of the source and the target domain to better generalize to the target domain. Different from these model-based approaches, we directly tackle the optimization issue from a meta-learning perspective.

2.3 Meta-Learning

Meta-learning or learning-to-learn, which can date back to some early works [Naik and Mamhomme, 1992], has recently attracted extensive attentions. A fundamental problem is “fast adaptation to new and limited observation data”. In pursuing this problem, there are three categories of meta-learning methods:

Metric-based. The idea is to learn a metric space and then use it to compare low-resource testing samples to rich training samples. The representative works in this category include Siamese Network [Koch et al., 2015], Matching Network [Vinyals et al., 2016], Memory-augmented Neural Network (MANN) [Santoro et al., 2016], Prototype Net [Snell et al., 2017], and Relation Network [Sung et al., 2018].

Model-based. The idea is to use an additional meta-learner to learn to update the original learner with a few training examples. [Andrychowicz et al., 2016] developed a meta-learner based on LSTMs. Hypernetwork [Ha et al., 2016], MetaNet [Munkhdalai and Yu, 2017], and TCML [Mishra et al., 2017] also learn a separate set of representations for fast model adaptation. [Ravi and Larochelle, 2017] proposed an LSTM-based meta-learner to learn the optimization algorithm (gradients) used to train the original network.

Optimization-based. The optimization algorithm itself can be designed in a way that favors fast adaption. Model-agnostic meta-learning (MAML, [Finn et al., 2017; Yoon et al., 2018; Gu et al., 2018]) achieved state-of-the-art performance by directly optimizing the gradient towards a good parameter initialization for easy fine-tuning on low-resource scenarios. It introduces no additional architectures nor parameters. Reptile [Nichol and Schulman, 2018] is similar to MAML with only first-order gradient. In this paper, we propose a generalized meta optimization method based on MAML to directly solve the intrinsic learning issues of low-resource NLG tasks.

3 Meta-Learning for Low-resource NLG

In this section, we describe the objective of fine-tuning a NLG model on a low-resource NLG task in Section 3.1. Then, we describe how our Meta-NLG algorithm encapsulates this objective into Meta NLG tasks and into the meta optimization algorithm to learn better low-resource NLG models.

3.1 Fine-tune a NLG Model

Suppose $f_\theta$ is the base NLG model parameterized by $\theta$, and we have an initial $\theta^*$ pre-trained with DA-utterance pairs $\mathcal{D}_s = \{(d_j, Y_j)\}_{j \in s}$ from a set $s$ of high-resource source tasks. When we adapt $f_\theta$ to some low-resource task $t$ with DA-utterance pairs $\mathcal{D}_t = (d_i, Y_i)$, the fine-tuning process on $\mathcal{D}_t$ can be formulated as follows:

$$\theta^* = \text{Adapt}(\mathcal{D}_t, \theta = \theta^*) = \arg\max_{\theta} \mathcal{L}_{\mathcal{D}_t}(f_\theta) = \arg\max_{\theta} \sum_{(d_i, Y_i) \in \mathcal{D}_t} \log P(Y_i|d_i; \theta)$$  (2)

The parameter $\theta^*$ will be used for initialization, and the model is further updated by new observations $\mathcal{D}_t$. The size of $\mathcal{D}_t$ in low-resource NLG tasks is very small due to the high annotation cost, therefore, a good initialization parameter $\theta^*$ learned from high-resource source tasks is crucial for the adaptation performance on new low-resource NLG tasks.
3.2 Meta NLG Tasks

To learn a \( \theta^* \) that can be easily fine-tuned on new low-resource NLG tasks, the idea of our Meta-NLG algorithm is to repeatedly simulate auxiliary Meta NLG tasks from \( D_s \) to mimic the fine-tuning process in Eq.(2). Then, we treat each Meta NLG task as a single meta training sample/episode, and utilize the meta optimization objective in the next section to directly learn from them.

Therefore, the first step is to construct a set of auxiliary Meta NLG tasks \( \{T_1, \ldots, T_K\} \) to simulate the low-resource fine-tuning process. We construct a Meta NLG task \( T_i \) by:

\[
T_i = (D_{T_i}, D'_{T_i})
\]  

(3)

\( D_{T_i} \) and \( D'_{T_i} \) of each \( T_i \) are two independent subsets of DA-utterance pairs from high-resource source data \( D_s \). \( D_{T_i} \) and \( D'_{T_i} \) correspond to meta-train (support) and meta-test (query) sets of a typical meta-learning or few-shot learning setup, and \( T_i \) is often referred to as a training episode. This meta setup with both \( D_{T_i} \) and \( D'_{T_i} \) in one Meta NLG task allows our Meta-NLG algorithm to directly learn from different Meta NLG tasks. The usage of them will be elaborated later. Meta NLG tasks are constructed with two additional principles:

**Task Generalization.** To generalize to new NLG tasks, Meta NLG tasks follow the same modality as the target task. For example, if our target task is to adapt to DA-utterance pairs in a new domain, then DA-utterance pairs in each \( T_i \) are sampled from the same source domain. We also consider adapting to new DA types in later experiments. In this case, DA-utterance pairs in each \( T_i \) have the same DA type. This setting merges the goal of task generalization.

**Low-resource Adaptation.** To simulate the process of adapting to a low-resource NLG task, the sizes of both subsets \( D_{T_i} \) and \( D'_{T_i} \), especially \( D'_{T_i} \), are set small. Therefore, when the model is updated on \( D'_{T_i} \) as a part of the later meta-learning steps, it only sees a small amount of samples in that task. This setup embeds the goal of low-resource adaptation.

3.3 Meta Training Objective

With the Meta NLG tasks defined above, we formulate the meta-learning objective of Meta-NLG as below:

\[
\theta^{Meta} = \text{MetaLearn}(T_1, \ldots, T_K) = \arg \max \theta \mathbb{E}_{D_{T_1} \ldots D_{T_K}} \mathcal{L}_{D'_{T_i}}(f_\theta) \tag{4}
\]

\[
\theta'_i = \text{Adapt}(D_{T_i}, \theta) = \theta - \alpha \nabla_{\theta} \mathcal{L}_{D'_{T_i}}(f_\theta) \tag{5}
\]

The optimization for each Meta NLG task \( T_i \) is computed on \( D'_{T_i} \) referring to \( D_{T_i} \). **Firstly**, the model parameter \( \theta \) to be optimized is updated on \( D_{T_i} \) by Eq.(5). This step mimics the process when \( f_\theta \) is adapted to a new low-resource NLG task \( T_i \) with low-resource observations \( D_{T_i} \). We need to note that Eq.(5) is an intermediate step, and it only provides an adapted parameter \( (\theta'_i) \) to our base model \( f \) to be optimized in each iteration. ** Afterwards**, the base model parameterized by the updated parameter \( (\theta'_i) \) is optimized on \( D'_{T_i} \) using the meta objective in Eq.(4). This meta-learning optimization objective directly optimizes the model towards generalizing to new low-resource NLG tasks by simulating the process repeatedly with Meta NLG tasks in Eq.(4).

The optimization of Eq.(4) can be derived in Eq.(6). It involves a standard first-order gradient \( \nabla_{\theta} \mathcal{L}_{D'_{T_i}}(f_\theta) \) as well as a gradient through another gradient \( \nabla_{\theta'} \mathcal{L}_{D'_{T_i}}(f_{\theta'}) \) .

\[
\theta \leftarrow \theta - \beta \sum_{i=1}^{K} \nabla_{\theta} \mathcal{L}_{D'_{T_i}}(f_{\theta'})
\]

\[
= \theta - \beta \sum_{i=1}^{K} \nabla_{\theta} \mathcal{L}_{D'_{T_i}}(f_{\theta'}) \cdot \nabla_{\theta} (\theta'_i)
\]

\[
= \theta - \beta \sum_{i=1}^{K} \nabla_{\theta} \mathcal{L}_{D'_{T_i}}(f_{\theta'}) \cdot (I - \alpha H_{\theta} (\mathcal{L}_{D'_{T_i}}(f_\theta)))
\]

(6)

To better understand the meta objective, we compare a standard multi-task learning (MTL) objective \( \theta^{MTL} = \arg \max_{\theta} \mathbb{E}_{D_{T_1} \ldots D_{T_K}} \mathcal{L}_{D'_{T_i}}(f_\theta) \) that learns from high-resource NLG tasks \( s_j \) without explicitly learning to adapt to low-resource NLG tasks. Figure 1 visually illustrates the differences with three high-resource source tasks \( \{s_1, s_2, s_3\} \) with optimal parameters \( \{\theta_1, \theta_2, \theta_3\} \) for each task. \( \theta^{MTL} \) is learned from individual DA-utterance pairs in \( \{D_{s_1}, D_{s_2}, D_{s_3}\} \), while Meta-NLG repeatedly constructs auxiliary Meta NLG tasks \( \{T_1, \ldots, T_K\} \) from \( \{D_{s_1}, D_{s_2}, D_{s_3}\} \) and learns \( \theta^{Meta} \) from them. As a result, \( \theta^{Meta} \) is closer to \( \theta_1 \) and \( \theta_2 \) (the optimal parameters of some new low-resource tasks, e.g, \( t_1 \) and \( t_2 \)) than \( \theta^{MTL} \). Algorithm 1 illustrates the process to learn \( \theta^{Meta} \) from \( D_s \). We note that batches are at the level of Meta NLG tasks, not DA-utterances pairs. Fine-tuning Meta-NLG on a new low-resource NLG task with annotated DA-utterance pairs \( D_t \) uses the same algorithm parameterized by \( (f_\theta, \theta^{Meta}, D_t, \alpha, \beta) \).
Algorithm 1 Meta-NLG($f_0$, $\theta_0$, $D_0$, $\alpha$, $\beta$)

Input: $f_0$, $\theta_0$, $D_0$, $\alpha$, $\beta$
Output: $\theta_{\text{Meta}}$

1: Initialize $\theta = \theta_0$
2: while $\theta$ not converge do
3: Simulate a batch of Meta NLG tasks $\{T_i = (D_{T_i}, D'_{T_i})\}_{i=1}^K$
4: for $i = 1$ to $K$ do
5: Compute $\theta_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{D_{T_i}}(f_0)$ in Eq.(5)
6: end for
7: Meta update $\theta \leftarrow \theta - \beta \nabla_{\theta} \mathcal{L}_{D'_{T_i}}(f_{\theta_i})$ in Eq.(6)
8: end while
9: Return $\theta_{\text{Meta}} = \theta$

4 Experiment

4.1 Baselines and Model Settings

We utilized the well-recognized semantically conditioned LSTM (SCLSTM [Wen et al., 2015c]) as the base model $f_0$

We used the default setting of hyperparameters ($n_{\text{layer}} = 1$, $\text{hidden}_{\text{size}} = 100$, dropout $= 0.25$, $\text{clip} = 0.5$, beam_width $= 5$). We implemented Meta-NLG based on the PyTorch SCLSTM implementation from [Budzianowski et al., 2018].

As Meta-NLG is model-agnostic, it is applicable to many other NLG models.

We included different model settings as baseline:

- Scratch-NLG: Train $f_0$ with only low-resource target task data, ignoring all high-resource source task data.
- MTL-NLG: Train $f_0$ using a multi-task learning paradigm on source task data, then fine-tune on the low-resource target task.
- Zero-NLG: Train $f_0$ using multi-task learning (MTL) with source task data, then directly test on a target task without a fine-tuning step. This corresponds to a zero-shot learning scenario.
- Supervised-NLG: Train $f_0$ using MTL with full access to high-resource data from both source and target tasks. Its performance serves an upper bound using multi-task learning without the low-resource restriction.
- Meta-NLG (proposed): Train $f_0$ using Algorithm 1 on source task data, then fine-tune on the low-resource target task.

For Meta-NLG, we set batch size to 50, and $\alpha = 0.1$ and $\beta = 0.001$. A single inner gradient update is used per meta update with Adam [Kingma and Ba, 2014]. The size of a Meta NLG task is set to 400 with 200 samples assigned to $D_{T_i}$ and $D'_{T_i}$. The maximum number of epochs is set to 100 during training and fine-tuning, and early-stop is conducted on a small validation set with size 200. The model is then evaluated on other DA-utterance pairs in the target task.

As in earlier NLG researches, we use the BLEU-4 score [Papineni et al., 2002] and the slot error rate (ERR) as evaluation metrics. ERR is computed by the ratio of the sum of the number of missing and redundant slots in a generated utterance divided by the total number of slots in the DA.

We randomly sampled target low-resource task five times for each experiment and reported the average score.

4.2 MultiWoz Dataset for NLG

We used a recently released large-scale multi-domain dialog dataset (MultiWOZ, [Budzianowski et al., 2018]). It is a proper benchmark for evaluating NLG components due to its domain complexity and rich linguistic variations. A visualization of DA types in different domains are given in Figure 2, and slots in different domains are summarized in Table 1. The average utterance length is 15.12, and almost 60% of utterances have more than one dialogue act types or domains. 69,607 annotated utterances are used, with 55,026, 7,291, 7,290 for training, validation, and testing respectively.

4.3 Domain Adaptation

In this section, we tested when a NLG model is adapted to various low-resource language domains. Experiment follows a leave-one-out setup by leaving one target domain for adaptation, while using the remainder domains for training. A target domain is a near-domain if it only contains domain-specific slots compared to the remainder domains. In contrast, a target domain containing both domain-specific DA types and slots is considered as a far-domain. According to Figure 2 and Table 1, “Attraction”, “Hotel”, “Restaurant”, and “Taxi”, are near-domains, while “Booking” and “Train” are far-domains. Adapting to near-domains requires to capture unseen slots, while adapting to far-domains additionally requires to learn new language patterns. Adaptation size is the number of DA-utterance pairs in the target domain to fine-tune the NLG model. To test different low-resource degrees, we considered different adaptation sizes (1,000, 500, 200).
Near-domain Adaptation. Results of adapting to two near-domains (“Attraction” and “Hotel”) are presented in Table 2. Other two near-domains (“Restaurant”, and “Taxi”) are simpler, therefore, they are not included due to page limit. Several observations can be noted. First, Using only source or target domain samples does not produce competitive performance. Using only source domain samples (Zero-NLG) performs the worst. It obtains very low BLEU-4 scores, indicating that the sentences generated do not match the linguistic patterns in the target domain. Using only low-resource target domain samples (Scratch-NLG) performs slightly better, yet still much worse than MTL-NLG and Meta-NLG. Second, Meta-NLG shows a very strong performance for this near-domain adaptation setting. It consistently outperforms MTL-NLG and other methods with very remarkable margins in different metrics and adaptation sizes. More importantly, it even works better than Supervised-NLG which is trained on high-resource samples in the target domain. Third, Meta-NLG is particularly strong in performance when the adaptation size is small. As the adaptation size decreases from 1,000 to 200, the performance of Scratch-NLG and MTL-NLG drops quickly, while Meta-NLG performs stably well. Both BLEU-4 and ERR even increase in “Hotel” domain when the adaptation size decreases from 500 to 200.

Far-domain Adaptation. Results of adapting to two far-domains (“Booking” and “Train”) are presented in Table 3. Again, we can see that Meta-NLG shows very strong performance on both far-domains with different adaptation sizes. Similar observations can be made as in the previous near-domain adaptation experiments. Because far-domain adaptation is more challenging, Meta-NLG does not outperform Supervised-NLG, and the performance of Meta-NLG drops more obviously as the adaptation size decreases. Noticeably, “Train” is more difficult than “Booking” as the former contains more slots, some of which can only be inferred from the smallest “Taxi” domain. The improvement margin of Meta-NLG over MTL-NLG and other methods is larger on the more difficult “Train” domain than on the “Booking” domain.

Table 2: Results for near-domain adaptation with different adaptation sizes. Bold numbers highlight the best results except Supervised-NLG.

Table 3: Results for far-domain adaptation with different adaptation sizes. Bold numbers highlight the best results except Supervised-NLG.

Table 4: Results for adapting to new DA type “Book” and “Recommend” with adaptation size 500.
NLG as fine-tuning epoch proceeds on the most challenging “Train” domain. The effect of meta-learning for low-resource NLG can be observed by comparing the two solid curves against the corresponding dashed curves. First, Meta-NLG adapts faster than MTL-NLG. We can see that the ERR of Meta-NLG (red-solid) decreases much more rapidly than that of MTL-NLG (red-dashed), and the BLEU-4 score of Meta-NLG (purple-solid) increases more quickly. The optimal BLEU-4 and ERR that MTL-NLG converges to can be obtained by Meta-NLG within 10 epochs. Second, Meta-NLG adapts better than MTL-NLG. As it can be seen, Meta-NLG achieves a much lower ERR and a higher BLEU-4 score when it converges, indicating that it found a better θ of the base NLG model to generalize to the low-resource target domain.

4.6 Manual Evaluation
To better evaluate the quality of the generated utterances, we performed manual evaluation.

Metrics. Given a DA and a reference utterance in a low-resource target domain with adaptation size 500, two responses generated by Meta-NLG and MTL-NLG were presented to three human annotators to score each of them in terms of informativeness and naturalness (rating out of 3), and also indicate their pairwise preferences (Win-Tie-Lose) on Meta-NLG against MTL-NLG. Informativeness is defined as whether the generated utterance captures all the information, including multiple slots and probably multiple DA types, specified in the DA. Naturalness measures whether the utterance is plausibly generated by a human.

Annotation Statistics. Cases with identical utterances generated by two models were filtered out. We obtained in total 600 annotations on each individual metric for each target domain. We calculated the Fleiss’ kappa [Fleiss, 1971] to measure inter-rater consistency. The overall Fleiss’ kappa values for informativeness and naturalness are 0.475 and 0.562, indicating “Moderate Agreement”, and 0.637 for pairwise preferences, indicating “Substantial Agreement”.

Results. Scores of informativeness and naturalness are presented in Table 6. Meta-NLG outscores MTL-NLG in terms of both metrics on all four domains. Overall, Meta-NLG received significantly (two-tailed t-test, p < 0.0005) higher scores than MTL-NLG. Results for pairwise preferences are summarized in Table 7. Even though there are certain amount of cases where the utterances generated by different models.

<table>
<thead>
<tr>
<th>Reference</th>
<th>The travel time is [79 minutes] and the cost is [17.60 pounds], shall I book for you?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTL-NLG</td>
<td>there is a train that leaves at [slot-train-leave] and arrives at [slot-train-arrive]. would you like me to book it for you? [missed: Ticket: Time; redundant: Leave. arrive]</td>
</tr>
<tr>
<td>Meta-NLG</td>
<td>the travel time is [79 minutes] and the price is [17.60 pounds], would you like me to book it for you? [correct]</td>
</tr>
</tbody>
</table>

Table 5: Sampled generated sentences when considering “Train” as the target domain with adaptation size 500. * indicates a domain-specific DA type, † indicates a domain-specific slot, and ‡ indicates a rare slot that can only be inferred from the smallest “Taxi” domain.

<table>
<thead>
<tr>
<th>Attraction</th>
<th>Hotel</th>
<th>Booking</th>
<th>Train</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meta-NLG</td>
<td>2.91</td>
<td>2.90</td>
<td>2.84</td>
</tr>
<tr>
<td>MTL-NLG</td>
<td>2.70</td>
<td>2.87</td>
<td>2.57</td>
</tr>
</tbody>
</table>

Table 6: Manual evaluation scores of informativeness (inf.), and naturalness (nat.) on four target low-resource domains.

<table>
<thead>
<tr>
<th>Attraction</th>
<th>Hotel</th>
<th>Booking</th>
<th>Train</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meta-NLG</td>
<td>50.2%</td>
<td>53.3%</td>
<td>40.1%</td>
</tr>
<tr>
<td>MTL-NLG</td>
<td>42.8%</td>
<td>42.3%</td>
<td>46.2%</td>
</tr>
</tbody>
</table>

Table 7: Pairwise preferences (Meta-NLG vs. MTL-NLG) on four target low-resource domains.

are nearly the same (Tie) to annotators, Meta-NLG is overall significantly preferred over MTL-NLG (two-tailed t-test, p < 0.0001) across different target domains.

4.7 Case Study
Table 5 shows two examples in the “Train” domain. The first sample shows that MTL-NLG fails to generate the domain-specific slot “Ticket”, instead, it mistakenly generates slots (“Leave” and “Arrive”) that are frequently observed in the low-resource adaptation set. In the second example, MTL-NLG failed to generate the domain-specific slot ‘Id’ and another rare slot “Dest”, while Meta-NLG succeeded both.

5 Conclusion
We propose a generalized optimization-based meta-learning approach Meta-NLG for low-resource NLG tasks. Meta-NLG utilizes Meta NLG tasks and a meta-learning optimization procedure based on MAML. Extensive experiments on a new benchmark dataset (MultiWoZ) show that Meta-NLG significantly outperforms other training procedures, indicating that it adapts fast and well to new low-resource settings. Our work may inspire researchers to use similar optimization techniques for building more robust and scalable NLG components in task-oriented dialog systems.

Acknowledgments
This work was jointly supported by the National Key R&D Program of China (Grant No. 2018YFC0830200) and the National Science Foundation of China (Grant No.61876096/61332007). We would also like to thank Prof. Xiaoyan Zhu for her unreserved support.
References


