	80			400			800			1600		
Model	Inform	Success	BLEU	Inform	Success	BLEU	Inform	Success	BLEU	Inform	Success	BLEU
SOLOIST	58.4	35.3	**10.58**	69.3	52.3	**11.8**	69.9	51.9	**14.6**	74	60.1	**15.24**
MAML	62.09	38.36	9.96	72.31	52.91	10.87	74.78	57.71	11.29	76.73	60.61	11.99
DAST	**62.70**	**38.68**	9.49	**74.52**	**54.45**	11.08	**75.92**	**57.72**	11.52	**77.95**	**60.87**	11.97

Table 4: The average performance over four domains, "Attraction", "Restaurant", "Train" and"Hotel", with 80/400/800/1600 dialogs for adaptation. DAST consistently achieves the best performance in the metrics of Inform rate and Success rate, with increasing the amount of adaptation data

tion, we find that the gap between SOLOIST and our methods reduces with an increasing amount of adaptation data. This is because, with enough data, the student model can learn the new domain well without the meta-teacher's guidance. Therefore, the influence of the meta-teacher declines as the number of adaptation dialogs increases. Table 3 describes the average performance in all the target domains from the Schema-Guided dataset. Our method achieves better performance compared to the MAML baseline in all metrics, suggesting that our method can generalize to different multi-domain dialog datasets.

Case Study and Visualization

Figure 3 lists four example sentences in the restaurant domain from the MultiWOZ dataset, along with their weights assigned by the meta-teacher model. To visualize the weight of each token, we color each token according to its corresponding weight. The larger the weight is, the darker the color is. Since we multiply the weights with token losses to update the student model, the absolute value of the weight can be considered as part of the learning rate. Therefore, we mainly focus on the relative values of weights within the same sentence. And the color intensity only suggests the relative value of weights in the same sentence. The first two sentences show that our meta-teacher model focuses more on the domain-related tokens like "area" and delexicalized slots such as "[value_area]". One possible reason is that general tokens (like "there are" in the second sentence) have already been learned by the student model in source domains while tokens like "[value_area]" appear less frequently in the source domains. Since large weight amplifies the feedback of the token loss during back-propagation, larger weights for the domain-related tokens encourage the student model to focus on those tokens and quickly learn features of the new domain, leading to more efficient adaptation. In the third case, we find that the delexicalized slots like "[value_range]" and "[value_address]" attracts more attention than domain-related token "address". This is because the domain-related tokens are still possible to be found in other domains. For example, "address" exists in five domains in the MultiWOZ dataset. The last sentence does not contain any domain-specific tokens. Hence, the token weights are close to each other. In this case, the weights do not make much impacts on updating model parameters.

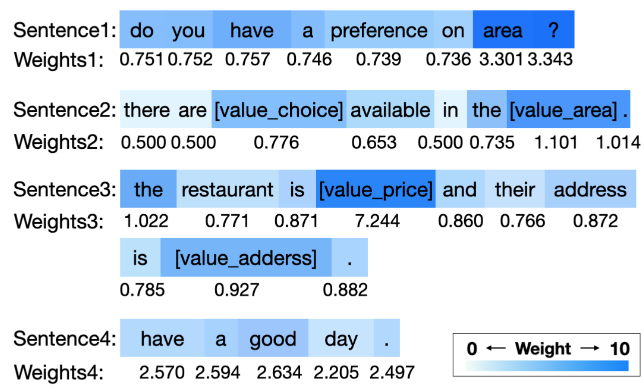

Figure 3: Visualizing weights corresponding to different tokens with different color intensities. The darker the color is, the larger the corresponding token's weight is.

Conclusion and Future work

We propose a domain adaptation method for low-resource task-oriented dialog systems, which incorporates a student-teacher architecture under the meta-learning setting. We present a transformer-based meta-teacher model, which learns to distinguish important tokens under different contexts across source domains during training. As for adaptation, the meta-teacher instructs the student dialog model to pay more attention to influential tokens by assigning weights to token losses, which improves the student model's adaptation performance. We evaluate our method on two popular human-human multi-domain datasets. The results demonstrate that our method reaches state-of-the-art performance in most task-related metrics, compared with MAML and SOLOIST. Since the meta-teacher is built to assign weights to a sequence of generated tokens, our method can be applied to other NLP tasks, such as machine translation and summarization. Furthermore, our meta-teacher model is compatible with other domain adaptation methods, such as MAML and pre-trained models.

In the future, we aim to extend our method in several directions. First, we plan to include the Success rate and Inform rate into the loss function of the meta-teacher model in a reinforcement learning setting. We believe directly optimize task success metric may lead to better performance. Another direction is to combine the meta-teacher model and pre-trained models to explore the compatibility, as well as replacing GRU-based student model with pre-trained

Cupiditate placeat eveniet deleniti est suscipit vel sequi officiis molestiae a, ipsa consectetur