TaskMix: Data Augmentation for Meta-Learning of Spoken Intent Understanding

Surya Kant Sahu
Skit.ai
The Learning Machines
`surya.oju@pm.me`

Abstract

Meta-Learning has emerged as a research direction to better transfer knowledge from related tasks to unseen but related tasks. However, Meta-Learning requires many training tasks to learn representations that transfer well to unseen tasks; otherwise, it leads to overfitting, and the performance degenerates to worse than Multi-task Learning. We show that a state-of-the-art data augmentation method worsens this problem of overfitting when the task diversity is low. We propose a simple method, TaskMix, which synthesizes new tasks by linearly interpolating existing tasks. We compare TaskMix against many baselines on an in-house multilingual intent classification dataset of N-Best ASR hypotheses derived from real-life human-machine telephony utterances and two datasets derived from MTOP. We show that TaskMix outperforms baselines, alleviates overfitting when task diversity is low, and does not degrade performance even when it is high.

1 Introduction

Deep learning has seen a meteoric rise in Speech and Language related applications, leading to large-scale applications of Voice-bots, Voice Assistants, Chatbots, etc., which aim to automate mundane tasks such as answering users' queries either in Spoken or textual modality. In many applications, users tend to code-switch or use borrowed words from other languages. A model trained for a particular language will not understand these borrowed words, and hence language-specific models are undesirable in such scenarios. On the other hand, a multilingual model can understand and reason what the user is speaking.

Due to the scale of the applications, data captured from various sources have different distributions or have different use-cases. Recently, Meta-Learning has emerged as a novel research direction that aims to leverage knowledge from diverse sets of data to learn a transferable initialization so that a low amount of training data is required to adapt to new datasets or tasks.

However, Meta-Learning requires a large number of training tasks, or else the model would overfit to the training tasks and would not generalize well to new tasks (Yao et al., 2021). In this work, we propose a novel Data Augmentation method, *TaskMix* for meta-learning problems, inspired by MixUp (Zhang et al., 2018). We investigate our proposed method against baselines such as MetaMix (Yao et al., 2021), Multitask-Learning, and vanilla Transfer Learning for multi-domain multi-lingual Spoken Intent Classification.

2 Preliminaries

In this section we describe the problem formulation and the prior work which we built upon.

2.1 Problem Formulation

Let $p(\mathcal{T})$ be a distribution over tasks from which training tasks $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{T-1}$ are sampled. The Meta-Learning objective is to learn a model with parameters θ such that θ quickly adapts to previously unseen tasks, which are assumed to be sampled from the same underlying distribution $p(\mathcal{T})$; for this paper, each task is a tuple $\mathcal{X}, \mathcal{Y} = \mathcal{T}$, where $\mathcal{X}$ is a set of N-Best hypotheses of utterances, and $\mathcal{Y}$ is a set of corresponding one-hot-encoded intent classes.

The number of classes in each $\mathcal{Y}$ may differ, and utterances from different $\mathcal{X}$ may be of different language or a different domain. This formulation is general and caters to real-life datasets.

Many meta-learning methods divide each training task into two disjoint sets: support $\mathcal{X}^s, \mathcal{Y}^s$ and query $\mathcal{X}^q, \mathcal{Y}^q$. However, Bai et. al (Bai et al., 2021) have shown that a query-set is unnecessary for meta-learning. Hence, throughout this work, we do not split the meta-training tasks, i.e., $\mathcal{X}^s = \mathcal{X}^q = \mathcal{X}$ and $\mathcal{Y}^s = \mathcal{Y}^q = \mathcal{Y}$
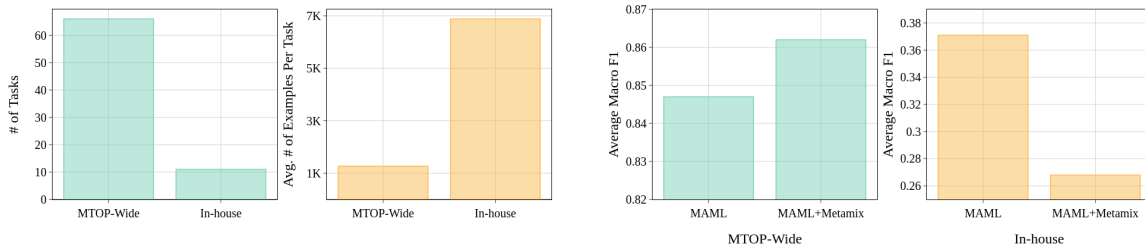
Figure 1: (Left) Statistics of the two datasets used in this paper. MTOP-Wide has a high #tasks and a low mean #examples per task; our In-house dataset has low #tasks, but a high mean #examples per task. (Right) Average Macro F1 scores of Model-Agnostic Meta-Learning (MAML) and MAML+MetaMix on both datasets. MetaMix is beneficial for MTOP-Long due to low mean #examples per task, whereas MetaMix worsens the performance our in-house dataset where the mean #examples per task is high.

Algorithm 1 MAML Update, $MetaTrain()$

Require: α: Learning rate for the inner loop.
Require: β: Learning rate for the outer loop.
Require: n: Iterations for the inner loop.
Require: $\mathcal{L}(t, \phi)$: Loss function for task t w.r.t. ϕ
1: **for** $\mathcal{T}_i \sim p(\mathcal{T})$ **do**   ▷ Sample from support set
2:     $\theta_i \leftarrow \theta$ ▷ Copy weights
3:     **for** $j = 1$ to n **do**
4:         Evaluate $\nabla_\theta \mathcal{L}(\mathcal{T}_i^s, \theta)$
5:         $\theta_i \leftarrow \theta_i - \alpha \nabla_\theta \mathcal{L}(\mathcal{T}_i^s, \theta)$
6:     **end for**
7: **end for**
8: $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i^q \sim p(\mathcal{T})} \mathcal{L}(\mathcal{T}_i^q, \theta_i)$ ▷ Update using query set

---

2.2 Model-Agnostic Meta-Learning

MAML (Finn et al., 2017) learns the meta-parameters θ by first, optimizing for multiple steps on a specific task $\mathcal{T}_i$, yielding θ_i which is the optimal task-specific parameters. This is done for each meta-training task $\mathcal{T}_i \sim p(\mathcal{T})$. Secondly, The loss on the held-out query set is computed, which is back-propagated through the computation graph through each task. Finally, we update θ such that θ can be quickly be adapted to each θ_i.

The procedure is outlined in Algorithm 1.

The authors argued that the held-out query set, which isn't used in the inner-loop optimization, prevents the overfitting of task-specific parameters θ_i and hence improves generalization of meta-parameters θ to new and unknown tasks.

However, (Bai et al., 2021) showed that splitting meta-training tasks into the disjoint query and support sets performs inferior to not splitting at all. Following these results, we do not split and sample

data from the same set for inner and outer loops.

2.3 MixUp

MixUp (Zhang et al., 2018) is a data augmentation technique that synthesizes new datapoints by linearly combining random datapoints in the training set, encouraging a simple, linear behavior between training examples, improving generalization and robustness to noise. The interpolation parameter λ is sampled randomly from the Beta distribution at each training step. As mixing sequences of discrete tokens, such as sentences, is not possible, following (Sun et al., 2020), we only mix the output features of the transformer model.

MetaMix uses MixUp to intra-task datapoints, creating new datapoints within the same task. Whereas our proposed method, TaskMix, extends MixUp to cross-task datapoints, creating *new meta-training tasks*.

2.4 MetaMix

MetaMix (Yao et al., 2021) is an application of MixUp to the meta-learning setting. MetaMix encourages generalization within tasks by combining query-set datapoints. Fig. 2 illustrates how *MAML+MetaMix* differs from *MAML*. MetaMix introduces an additional gradient for each task by mixing random datapoints within each task. MetaMix is a data augmentation method for Meta-Learning where MixUp is applied to random pairs of datapoints *within a batch of query set datapoints* of each task.
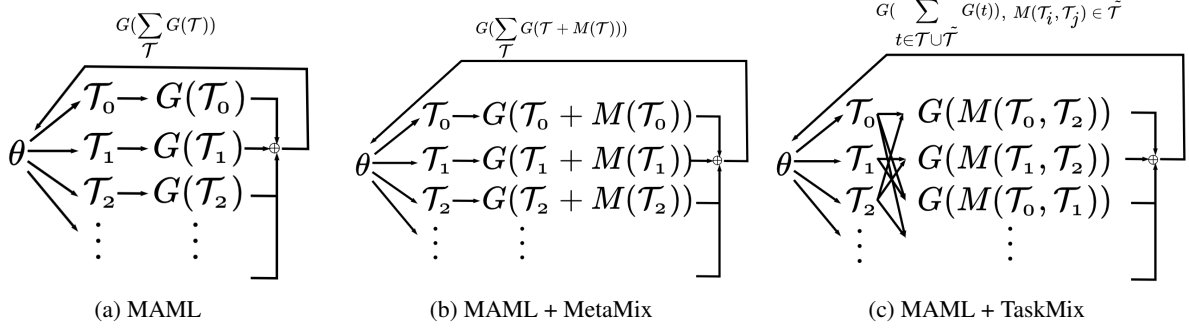
Figure 2: Illustration of variants of MAML, including our proposed method, TaskMix. Here, $M(\mathcal{T})$ denotes mixing of datapoints within $\mathcal{T}$; $M(\mathcal{T}_i, \mathcal{T}_j)$ denotes mixing of tasks $\mathcal{T}_i$ and $\mathcal{T}_j$; and G denotes the gradient operator. MetaMix mixes random pairs of datapoints within each task. TaskMix mixes random pairs of tasks at each iteration.

3 TaskMix

3.1 Motivation

By virtue of meta-learning, θ learns features that transfer well across tasks (Raghu et al., 2020), which requires a large number of meta-training tasks, and most datasets on which studies on Meta-Learning literature use datasets which have a very high number of tasks and low average number of training examples per task. We make the following observations:

- MetaMix increases the effective number of datapoints *within each task*, i.e., increasing the mean #examples per task.

- MetaMix does not change the effective number of tasks.

- From Fig. 1, we infer that in MTOP-Wide dataset, where the mean #examples per task is low, MetaMix is very beneficial; however, in our In-house dataset, MetaMix deteriorates performance as the mean #examples per task is already very high.

- Similar to many real-life multi-domain settings, our In-house dataset has a small number of tasks.

To this end, we propose a simple data-augmentation method, *TaskMix*, to increase the effective number of tasks used in meta-learning.

3.2 Method

We propose a simple method, *TaskMix*, to overcome the low task-diversity problem. First, we sample support and query set batches from all tasks; we then sample N pairs of task indices $\mathbf{I}, \mathbf{J}$ uniformly

---

Algorithm 2 TaskMix

Require: η : Beta distribution parameter
Require: $mix(a, b, \lambda) = \lambda a + (1 - \lambda)b$
Require: N : Number of new tasks to generate.
1: **while** not converged **do**
2:     **for** $t = 0$ to $T - 1$ **do**
3:         $x_t^q \sim \mathcal{X}_t^q, y_t^q \sim \mathcal{Y}_t^q$
4:         $x_t^s \sim \mathcal{X}_t^s, y_t^s \sim \mathcal{Y}_t^s$
5:     **end for**
6:     $\mathbf{I}, \mathbf{J} \sim \mathbf{U}^N(0, T - 1)$
7:     **for** $i \in \mathbf{I}, j \in \mathbf{J}, n = 0$ to $N - 1$ **do**
8:         $\lambda \sim Beta(\eta, \eta)$
9:         $\tilde{x}_n^q = mix(x_i^q, x_j^q, \lambda)$
10:        $\tilde{y}_n^q = mix(y_i^q, y_j^q, \lambda)$
11:        $\tilde{x}_n^s = mix(x_i^s, x_j^s, \lambda)$
12:        $\tilde{y}_n^s = mix(y_i^s, y_j^s, \lambda)$
13:     **end for**
14:     $MetaTrain()$
15: **end while**

---

in the range $[0, T - 1]$. For each selected task pair, we sample the interpolation parameter λ from the Beta distribution with parameters (η, η); and then mix the training examples from the support and query sets, resulting in a new synthetic task $\tilde{\mathcal{T}}_n$. Finally, we train with vanilla MAML; however, we train on the new task set $\mathcal{T} \cup \tilde{\mathcal{T}}$. Algorithm 2 describes this procedure.

TaskMix interpolates between batches of data-points of random meta-training tasks. In essence, TaskMix encourages generalization across tasks by synthesizing new tasks, while MetaMix encourages generalization within each task by synthesizing new datapoints within the task. We emphasize that TaskMix increases the effective number of tasks, whereas MetaMix increases the effective number

of datapoints within each task. We illustrate this difference in Fig. 2. We note that TaskMix and MetaMix are orthogonal, and *both methods can be used at the same time*.

TaskMix introduces only one additional hyperparameter, i.e., the number of synthetic tasks N. We found that results are largely insensitive to N if $N > T$, but performance rapidly degrades to the performance of MAML if $N < T$, hence we set $N = T$ for all experiments. We recover MAML if we set $N = 0$.

4 Experiments

This section presents empirical results on two multilingual and multi-domain datasets. For choice of hyperparamters and other experimental details, please refer to the Appendix.

4.1 Methods and Baselines

We use the N-Best-ASR Transformer (Ganesan et al., 2021) convention of concatenating N-Best ASR transcription hypotheses and then feed the concatenated text to XLM-RoBERTa (Conneau et al., 2020) feature extractor. We use the "base" configuration of pretrained XLM-RoBERTa to extract 768-dimensional vectors of each example for each task. The extracted features are inputs to a *neck*, which is a stack of Linear-Parametric ReLU layers. We chose XLM-RoBERTa as the feature extractor as it is trained on large corpora of multilingual text.

We now describe the baselines used in the experiments:

- **Multitask Learning (MTL):** we learn a different *linear head* for each meta-training task, and discard these heads after training, and initialize a new head for each meta-testing task.

- **Vanilla Transfer:** we discard all meta-training tasks and finetune directly to each meta-testing task.

- **MAML:** we append a linear layer with the max number of classes in the respective datasets.

4.2 Datasets

We briefly summarize the datasets used in this paper. Various statistics relating to the datasets are in Table 1.

Dataset	#Tasks	Mean #Classes	Mean #Examples Per Task
In-house	11	7.73	6884
MTOP-Long	11	2.82	7615
MTOP-Wide	66	2.17	1269

Table 1: Various statistics of datasets used in this paper.

- **In-house** dataset is constructed by collecting and automatically transcribing phone calls from various customer-call centers (varying domains, such as restaurants, airlines, banking, etc.) across 3 countries and with conversations comprising at least 3 languages with users speaking with borrowed words, code-switching, etc. Multiple human annotators manually label the intent for each user turn (consisting of 5-Best ASR hypotheses) in a conversation. The resulting dataset contains about 70K utterances across 11 tasks, grouped into 7 meta-training and 4 meta-testing tasks. We grouped the meta-training and meta-testing tasks chronologically, i.e., the oldest 7 tasks were designated as the meta-training tasks and the rest as meta-testing tasks. We use this setup to have as low an application gap as possible.

- **MTOP-Wide** (Li et al., 2021) contains over 100K utterances, (which we treat as 1-best hypotheses) from 6 languages across 11 domains. We divide the MTOP dataset by grouping examples from distinct domains and languages, resulting in 66 subsets. We further group these subsets into 54 meta-training and 14 meta-testing tasks. We only keep examples for which the class frequency is at least 50. We create this dataset to have a high task diversity but low average #examples per task.

- **MTOP-Long** (Li et al., 2021) We divide the MTOP dataset by grouping examples from distinct domains resulting in 11 subsets. We further group these subsets into 7 meta-training and 4 meta-testing tasks. We only keep examples for which the class frequency is at least 20. We create this dataset to have a low task diversity but high average #examples per task.

Method	Average Macro F1
MTL	0.320 ± 0.004
Vanilla Transfer	0.321 ± 0.007
MAML	0.361 ± 0.021
MAML+MetaMix	0.265 ± 0.006
MAML+TaskMix	0.370 ± 0.023
MAML+MetaMix+TaskMix	**0.441 ± 0.002**

Table 2: Results on our In-house dataset. We observe that TaskMix yields a significant performance boost. MAML+MetaMix degrades performance to worse than MAML.

Method	Average Macro F1
MTL	0.439 ± 0.022
Vanilla Transfer	0.446 ± 0.014
MAML	0.442 ± 0.002
MAML+MetaMix	0.450 ± 0.011
MAML+TaskMix	**0.462 ± 0.012**
MAML+MetaMix+TaskMix	0.421 ± 0.008

Table 3: Results on the MTOP-Long (Li et al., 2021) dataset. MAML+TaskMix out-performs other baselines.

Method	Average Macro F1
MTL	0.826 ± 0.018
Vanilla Transfer	0.804 ± 0.003
MAML	0.847 ± 0.006
MAML+MetaMix	**0.862 ± 0.006**
MAML+TaskMix	0.856 ± 0.003
MAML+MetaMix+TaskMix	0.861 ± 0.017

Table 4: Results on the MTOP-Wide (Li et al., 2021) dataset. MetaMix is beneficial and TaskMix doesn't negatively affect performance (compared to MAML) even when task diversity is high.

- For the In-house dataset, MetaMix degrades performance to be comparable to vanilla-transfer, i.e., almost no gain from meta-training tasks. We infer that MetaMix makes the model overfit on meta-training tasks, as the number of examples-per-task is already very high.

- In any of the datasets, TaskMix *doesn't degrade* the performance of MAML.

- For MTOP-Wide, TaskMix only has a slight performance boost compared to other baselines, suggesting that TaskMix is not useful if the number of tasks is already high.

We interestingly find that MAML+MetaMix+TaskMix is the worst performing method for MTOP-Long. However, TaskMix is beneficial when used on its own. We leave studying the interaction between MetaMix and TaskMix for future work.

5 Conclusion

In this paper, we propose a novel data-augmentation method, TaskMix, to alleviate the problem of overfitting in Meta-learning datasets when the task diversity is too low. Through experiments on two multilingual, multi-domain intent classification datasets, MetaMix could worsen the overfitting problem when the task diversity is low, whereas TaskMix is beneficial in such cases.

6 Acknowledgement

4.3 Evaluation

As all tasks across all datasets are highly imbalanced, we use the Macro F1 score to weigh all classes equally.

All tasks are grouped into meta-training and meta-testing sets; each task is split into "support" and "test" sets. For modeling, we first train on the meta-training tasks, then use the same weights to fine-tune on the meta-testing tasks, and then compute Macro F1 scores for each meta-testing task. We then compute the mean of Macro F1 scores across all meta-testing tasks. We denote this metric as *Average Macro F1*. Finally, we report the mean and standard deviation of Average Macro F1 scores across three independent trials with different seeds.

4.4 Results and Discussion

We make the following key observations from Tables 2, 3, and 4:

- TaskMix improves performance on "long" datasets i.e., on In-house and MTOP-Long where the #meta-training tasks are very low and # examples per task is high.

References

Yu Bai, Minshuo Chen, Pan Zhou, Tuo Zhao, J. Lee, Sham M. Kakade, Haiquan Wang, and Caiming Xiong. 2021. How important is the train-validation split in meta-learning? In *ICML*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

Chelsea Finn, P. Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Karthik Ganesan, Pakhi Bamdev, B. Jaivarsan, Amresh Venugopal, and Abhinav Tushar. 2021. N-best asr transformer: Enhancing slu performance using multiple asr hypotheses. *ArXiv*, abs/2106.06519.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *ArXiv*, abs/2008.09335.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. 2020. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *ArXiv*, abs/1909.09157.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S. Yu, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for nlp tasks. In *COLING*.

Huaxiu Yao, Long-Kai Huang, Linjun Zhang, Ying Wei, Li Tian, James Zou, Junzhou Huang, and Zhenhui () Li. 2021. Improving generalization in meta-learning via task augmentation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11887–11897. PMLR.

Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. *ArXiv*, abs/1710.09412.