Low-Resource Machine Translation through the Lens of Personalized Federated Learning

Anonymous ACL submission

Abstract

We present a new approach based on the Personalized Federated Learning algorithm `MeritFed` that can be applied to Natural Language Tasks with heterogeneous data. We evaluate it on the Low-Resource Machine Translation task, using the dataset from the Large-Scale Multilingual Machine Translation Shared Task (Small Track #2) and the subset of Sami languages from the multilingual benchmark for Finno-Ugric languages. In addition to its effectiveness, `MeritFed` is also highly interpretable, as it can be applied to track the impact of each language used for training. Our analysis reveals that target dataset size affects weight distribution across auxiliary languages, that unrelated languages do not interfere with the training, and auxiliary optimizer parameters have minimal impact. Our approach is easy to apply with a few lines of code, and we provide scripts for reproducing the experiments.[1]

1 Introduction

While 7,000+ languages are currently in use worldwide, most existing Natural Language Processing (NLP) tasks and Large Language Models (LLMs) cover at most 500 of them (Logacheva et al., 2020; ImaniGooghari et al., 2023; Lin et al., 2024). Many languages still possess low amount of resources, and a lot of NLP tasks for such languages remain unsolved. These facts indicate the difficulty and non-triviality of using LLMs that typically require large amounts of data. A popular direction of approaching low-resource languages (LRLs) is Machine Translation: automatic translation between most of these low-resource languages to high-resource ones is more economically and socially motivated than developing language-specific systems (Ranathunga et al., 2023).

To solve the tasks for LRLs, a lot of studies employ the related languages or languages originating from the same geographical and historical background (ImaniGooghari et al., 2023; Da Dalt et al., 2024; Millour et al., 2024). Despite the positive effect, it usually requires empirical knowledge, and many guesses and trials of different approaches when choosing the best combination of languages used, the most suitable amount of data, and the best learning strategy (Hedderich et al., 2021).

New approach. To address these issues, we present our approach called `MeritFed` to train LLMs for the target language while multiple datasets for different languages are available. The key idea behind our method is inspired by Tupitsa et al. (2024), who focus on a specific (Personalized) Federated Learning formulation (Kairouz et al., 2021). We focus on exploring the underlying algorithmic techniques in application to heterogeneous datasets rather than the Distributed Training.

Our approach is more robust as it adjusts the impact of each language (*aggregation weights*) during training without any explicit inductive bias towards language relatedness. In particular, our strategy benefits from the updates from the "important" languages and tolerates the updates from the "not important" ones. This setup is extremely beneficial for the interpretability of the training process.

In this study, we primarily focus on low-resource languages. However, our approach can be applied to any similar task (not necessarily in NLP). The main requirement is to possess multiple heterogeneous input datasets, while the goal is to train the model suitable for some target data distribution.

Therefore, we apply the algorithm to the Machine Translation task using two datasets: the subset from the Large-Scale Multilingual Machine Translation Shared Task (Small Track #2) (Wenzek et al., 2021) and the subset of Sami languages from the multilingual benchmark for Finno-Ugric languages (Yankovskaya et al., 2023). To test the method effectively within our compute budget, we

[1] https://anonymous.4open.science/r/MeritFed_review-2D5B

focus our study on scenarios with one target language and the remaining languages as auxiliary languages. Our approach can be further applied to the datasets with several target languages and several translation directions.

Two research questions are addressed in this paper: (i) *"Can* `MeritFed` *improve the results of the multilingual or single language baselines using aggregation weights?"* and (ii) *"How do the target language weights and the weights of related and non-related languages change across training?"*.

The contributions of the paper are as follows:

- We present a new algorithmic framework for the training from heterogeneous input datasets and test it on the World Machine Translation Shared task on the Indonesian languages and Sami languages of the Finno-Ugric Machine Translation benchmark.
- We explore how languages interact with each other during training, as our approach allows measuring the impact (which language contributes more) at each training step.
- We perform an ablation study to analyze the effects of unrelated languages, training dataset size, and auxiliary `MeritFed` parameters.
- Under certain assumptions, we rigorously prove that the proposed method converges to some neighborhood of the solution.

2 Related Work

In this section, we discuss the existing methods for low-resource language NLP tasks, especially for low-resource machine translation (LRMT) (Haddow et al., 2022), and also give a brief overview of the existing methods in Personalized Federated Learning, and methods to estimate the impact of auxiliary data.

2.1 Low-Resource Machine Translation

Existing approaches for NLP tasks for LRLs usually fall into the following categories: supervised or unsupervised, single language training or multilingual training, continuous pre-training or finetuning, with or without data augmentation, balanced or imbalanced datasets (Hedderich et al., 2021; Wang et al., 2021; Krasadakis et al., 2024; Goyal et al., 2020). This list of categories is not extensive. However, they all aim to develop the best learning strategy given limited data.

In the following subsections, we discuss the methods developed or applied for the datasets on South East Asian Languages and Finno-Urgic benchmarks, the main targets of our research.

2.1.1 LRMT for South East Asian Languages

Several approaches have been developed to solve the Large-scale Multilingual Machine Translation task (Shared Task on WMT-21). The organizers (Wenzek et al., 2021) summarize all the used approaches and provide the FLORES model (Goyal et al., 2022) extended to 124 languages. Most of the participants, Yang et al. (2021); Budiwati et al. (2021); Liao et al. (2021), use a generic pre-trained multilingual models like DeltaLM (Ma et al., 2021) or FLORES (Goyal et al., 2022) and fine-tune it correspondingly with the vast collected parallel data, together with applying progressive learning and iterative back-translation. Sutawika and Cruz (2021) use a standard Seq2Seq Transformer model without any training or architecture tricks, relying mainly on the strength of the data preprocessing techniques and filtering.

Given our focus on a setup with very limited data and our available computational resources, we concentrate on evaluating our specific approach. Therefore, our results cannot be compared to the above-mentioned methods.

2.1.2 LRMT for Finno-Ugric languages

Regarding the Finno-Ugric languages, very few approaches are developed or tested on the benchmark. Tars et al. (2022) uses the standard M2M100 model (Fan et al., 2021) enhanced with the following steps: vocabulary extension in the tokenizer, data filtering, and preprocessing. Yankovskaya et al. (2023) improves previous results with back-translation and synthetic data as well as with the sampled high-resource language pairs to reduce catastrophic forgetting. Our models involve the same baselines; however, our training data consists of Sami languages (input) and Finish (output). Therefore, we also cannot compare the results directly to the above-mentioned methods.

2.2 Personalized Federated Learning

Federated Learning (FL) (Konecnỳ et al., 2016; McMahan et al., 2017) is a modern and rapidly developing part of Machine Learning, considering the training on the data distributed over multiple clients (Kairouz et al., 2021). In the standard scenario, the goal is to train one global model that

suits multiple clients, i.e., solve standard empirical risk minimization. In scenarios with heterogeneous data, the global model can show suboptimal results for particular clients, which necessitates considering Personalized Federated Learning (PFL) formulations to achieve better results on the client's data while getting benefits from collaboration with others.

In the training of LLMs for the target (low-resource) language using the data in multiple languages, the goal is quite similar: to achieve good results for the target language while getting benefits from the model updates for other available languages. Therefore, in our work, we adjust the algorithmic ideas from (Tupitsa et al., 2024) to the training of LLMs for low-resource languages.

There also exist multiple PFL formulations and methods for solving them with their own advantages and limitations, e.g., see (Fallah et al., 2020; Collins et al., 2021; Hanzely et al., 2020; Kulkarni et al., 2020; Wu and Wang, 2021). However, the works on PFL focus on different scenarios from our setup, i.e., they consider distributed training.

2.3 Impact of Auxilary Data

Many existing papers rely on auxiliary data, especially when the given dataset is too small. Schröder and Biemann (2020) automatically assesses the similarity of sequence tagging datasets to identify beneficial auxiliary data for Multi-Task Learning or Transfer Learning setups. Chen et al. (2022) propose a joint task and data scheduling model for auxiliary learning by creating a mapping from task, feature, and label information to the schedule in a parameter-efficient way.

Regarding LRMT, studies use the related languages when little data for the target language is given. One of the attempts to approach each language differently during training is made by Huo et al. (2024). They dynamically allocate parameters of an appropriate scale to each language direction based on the consistency between the gradient of the individual language and the average gradient. Millour et al. (2024); Da Dalt et al. (2024) show that datasets on closely related languages are highly beneficial for applying to the target low-resource language. ImaniGooghari et al. (2023) also investigate the positive effects of closely related languages on the Glot-500 model. They analyze the impact of related languages via continued pre-training and confirm better performance for languages with their language family or script present in training.

3 Methodology

General setup. We start with the description of the general problem formulation that our approach is suitable for. That is, we consider the scenario when $n \geq 1$ datasets $\{\boldsymbol{D}_i\}_{i=1}^n$ are available for training, and the goal is to train the model for some data distribution $\mathcal{D}$ using this collection of datasets. More precisely, we focus on the standard learning problem (Shalev-Shwartz and Ben-David, 2014): $\min_{x\in\mathbb{R}^d} f_{\mathcal{D}}(x)$, where $f_{\mathcal{D}} : \mathbb{R}^d \to \mathbb{R}$ is the expected loss computed for the data distribution $\mathcal{D}$, i.e., $f_{\mathcal{D}} := \mathbb{E}_{\xi\sim\mathcal{D}}[f_\xi(x)]$ with $f_\xi : \mathbb{R}^d \to \mathbb{R}$ being a loss on sample ξ and $\mathbb{E}_{\xi\sim\mathcal{D}}[\cdot]$ denoting an expectation w.r.t. ξ coming from the target distribution $\mathcal{D}$, and $x \in \mathbb{R}^d$ represents a vector of model parameters, i.e., weights of the network. In practice, data distribution $\mathcal{D}$ is typically unknown. Therefore, to approximate $f_{\mathcal{D}}(x)$, finite dataset $\widehat{\boldsymbol{D}}$ sampled from distribution $\mathcal{D}$ is used. Throughout the paper, we call this dataset the target one and denote the corresponding (empirical) loss as $f_{\widehat{\boldsymbol{D}}}(x)$. In addition, we assume that a collection of datasets $\{\boldsymbol{D}_i\}_{i=1}^n$ is available for the training.

We assume that $\boldsymbol{D}_1$ is sampled from the target distribution $\mathcal{D}$, and we make no assumptions on the other datasets. In particular, $\{\boldsymbol{D}_i\}_{i=2}^n$ can be arbitrary heterogeneous and different from $\boldsymbol{D}_1$ and $\widehat{\boldsymbol{D}}$. However, if some of the available datasets are sampled from distributions that are close to $\mathcal{D}$, they can be quite useful for the training. This idea serves as the main motivation behind our approach.

Algorithmic framework. To solve the described problem, we propose a generic algorithmic framework – `MeritFed` (see Algorithm 1) – inspired by `MeritFed-SGD` proposed by Tupitsa et al. (2024) for solving Personalized Federated Learning problems. `MeritFed` can be seen as a "wrapper" for an optimization method having update rule $x^{t+1} = \texttt{OptStep}(x^t, g(x^t), \gamma_t)$, where x^t represents the weights of the model after step t, $g(x^t)$ is the stochastic (mini-batched) gradient computed at x^t, and γ_t is the learning rate. For example, when the underlying method is Stochastic Gradient Descent (`SGD`) (Robbins and Monro, 1951), we have $\texttt{OptStep}(x^t, g(x^t), \gamma_t) = x^t - \gamma_t g(x^t)$ and Algorithm 1 reduces to `MeritFed-SGD` from (Tupitsa et al., 2024). However, we can apply `MeritFed` to the update rule of any stochastic first-order method, e.g., `Adam` (Kingma and Ba, 2015) and its variations, `AdaGrad` (Streeter and McMahan, 2010; Duchi et al., 2011), `RMSProp` (Hinton et al., 2012),

Algorithm 1 `MeritFed`: General Algorithmic Framework for Learning from Heterogeneous Data

Input: Number of steps T, starting point $x^0 \in \mathbb{R}^d$, stepsizes $\{\gamma_t\}_{t=1}^T$ ($\gamma_t > 0$), optimization update rule $\texttt{OptStep}(x, g, \gamma) : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$, datasets $\{\boldsymbol{D}_i\}_{i=1}^n$, target validation dataset $\widehat{\boldsymbol{D}}$
for $t = 0, 1, \ldots, T$ **do**
  for all $i = 1, \ldots, n$ **in parallel do**
    Compute stochastic gradient $g_i(x^t)$ from dataset $\boldsymbol{D}_i$
  end for
  $w^{t+1} \approx \arg\min_{w \in \Delta_1^n} f_{\widehat{\boldsymbol{D}}}\left(\texttt{OptStep}\left(x^t, \sum_{i=1}^n w_i g_i(x^t), \gamma_t\right)\right)$
  $x^{t+1} = \texttt{OptStep}\left(x^t, \sum_{i=1}^n w_i^{t+1} g_i(x^t), \gamma_t\right)$
end for

and other methods. In our experiments, we use `Adam` as $\texttt{OptStep}(x, g, \gamma)$.

In addition to the update rule $\texttt{OptStep}(x, g, \gamma)$, `MeritFed` takes n input datasets $\{\boldsymbol{D}_i\}_{i=1}^n$ and 1 target validation dataset $\widehat{\boldsymbol{D}}$. At each iteration, the method computes (mini-batched) stochastic gradient $g_i(x^t)$ using the corresponding dataset $\boldsymbol{D}_i$ for each $i = 1, \ldots, n$. Then, to construct the update direction, `MeritFed` searches appropriate aggregation weights $w^{t+1} = (w_1^{t+1}, \ldots, w_n^{t+1})^\top$ (see Line 6) and then makes a step $x^{t+1} = \texttt{OptStep}\left(x^t, \sum_{i=1}^n w_i^{t+1} g_i(x^t), \gamma_t\right)$ using the computed weighted average of the stochastic gradients. We emphasize that the choice of aggregation weights w^{t+1} is crucial: for example, if datasets $\{\boldsymbol{D}_i\}_{i=2}^n$ came from distributions significantly different from the target distribution $\mathcal{D}$ and we choose uniform weights, i.e., $w_1^{t+1} = \ldots = w_n^{t+1} = 1/n$, then the optimization step with the update vector $\sum_{i=1}^n w_i^{t+1} g_i(x^t)$ can be useless (on average) in terms of solving the target problem. Moreover, if some datasets came from distributions close to $\mathcal{D}$, it is natural to use the corresponding stochastic gradients with larger weights to benefit from them.

`MeritFed` addresses this issue in Line 6: the goal is to find aggregation weights $w^{t+1} \in \Delta_1^n$, where $\Delta_1^n := \{y \in \mathbb{R}^n \mid \sum_{i=1}^n y_i = 1,\ y_i \geq 0\ \forall\ i = 1, \ldots, n\}$ is the n-dimensional probability simplex, such that the loss $f_{\widehat{\boldsymbol{D}}}$ on the target validation dataset $\widehat{\boldsymbol{D}}$ is minimized after the step $\texttt{OptStep}\left(x^t, \sum_{i=1}^n w_i^{t+1} g_i(x^t), \gamma_t\right)$ that depends on w^{t+1}. If $\widehat{\boldsymbol{D}}$ is sufficiently large, then $f_{\widehat{\boldsymbol{D}}}$ can be seen as a good approximation of $f_{\mathcal{D}}$ (Shalev-Shwartz et al., 2009), and optimizing $f_{\widehat{\boldsymbol{D}}}$ leads to sufficiently good solution for $f_{\mathcal{D}}$. In other words, given stochastic gradients $g_i(x^t)$ computed from different datasets $\{\boldsymbol{D}_i\}_{i=1}^n$, `MeritFed` tries to find the best-weighted average of them to make an optimization step. Following Tupitsa et al. (2024), we apply several steps of Stochastic Mirror Descent (`SMD`) (Nemirovskij and Yudin, 1983) to solve the problem in Line 6 approximately.

Application to NLP. The described approach can be applied to the training of LLMs for LRLs. In this case, $\{\boldsymbol{D}_i\}_{i=1}^n$ correspond to the input datasets in n different languages. In particular, $\boldsymbol{D}_1$ is the training dataset for the target language[2] and $\widehat{\boldsymbol{D}}$ is the target validation dataset for the same language. The remaining datasets $\{\boldsymbol{D}_i\}_{i=2}^n$ are for other languages. Some of these languages can be related to the target one, but, in general, we allow the usage of datasets in significantly different languages as well: `MeritFed` automatically adjusts aggregation weights and assigns higher weights to more beneficial languages. Therefore, aggregation weights w^{t+1} can be used to measure the impact of selected languages on the model's training for the target language. In other words, we extend the training target language dataset and prevent drifting towards the solution for other languages.

4 Experiments

In this section, we apply the methodology to learn low-resource languages with the help of related languages. We also discuss the data used, the baselines, and the evaluation metrics.

4.1 Datasets

To test the developed method, we need to consider datasets with related languages that either belong to the same language family or are geographically

[2] One can interpret all possible texts in the target language as some distribution $\mathcal{D}$. In this interpretation, $\boldsymbol{D}_1$ can be seen as some dataset sampled from language $\mathcal{D}$.

Method	Inari Sami		Skolt Sami		South Sami		North Sami	
	Score	Steps	Score	Steps	Score	Steps	Score	Steps
CP_{All}	51.39 ± 0.05	30K	44.90 ± 0.12	25K	11.60 ± 0.29	23K	39.78 ± 0.08	69K
CP_{NoT}	50.14 ± 0.04	31K	43.40 ± 0.13	25K	11.09 ± 0.24	23K	39.30 ± 0.18	65K
`MeritFed`	**52.08 ± 0.01**	12K	**50.27 ± 0.17**	12K	**13.26 ± 0.17**	2.5K	38.526 ± 1.39	30K

Table 1: Mean SpBLEU scores and the number of steps required to reach them for baselines and `MeritFed` within Finno-Samic low-resource languages.

Method	Tagalog						Javanese					
	Small		Medium		Large		Small		Medium		Large	
	Score	Steps	Score	Steps	Score	Steps	Score	Steps	Score	Steps	Score	Steps
CP_{All}	29.24 ± 0.06	21K	30.99 ± 0.04	40K	33.89 ± 0.15	124K	19.43 ± 0.14	12K	20.05 ± 0.12	25K	20.97 ± 0.13	87K
CP_{NoT}	28.72 ± 0.16	15K	30.50 ± 0.12	42K	33.74 ± 0.19	129K	19.46 ± 0.12	12K	19.95 ± 0.12	25K	21.19 ± 0.09	89K
`MeritFed`	**29.73 ± 0.04**	14K	**31.42 ± 0.07**	14K	33.53 ± 0.27	47K	**19.74 ± 0.03**	2K	**20.23 ± 0.11**	3K	**21.44 ± 0.13**	8K

Table 2: Mean SpBLEU scores and the number of steps required to reach them for baselines and `MeritFed` within the different data sizes of Javanese and Tagalog languages.

related, which we expect to be "helpful" during the training procedure. Therefore, we select a subset from the Large-Scale Multilingual Machine Translation Shared Task (Small Track #2) (Wenzek et al., 2021) and the subset of Sami languages from the multilingual benchmark for Finno-Ugric languages (Yankovskaya et al., 2023). We describe each dataset in detail in the following paragraphs.

South East Asian languages Dataset. For the first round of experiments, we select one of the small tracks, Large-Scale Multilingual Machine Translation Shared Task, comprising translation pairs between fairly related languages and English and not requiring substantial computational resources at training time. We stick to Javanese, Indonesian, Malay, Tagalog, and Tamil as input languages and English as output. As target languages, we utilize Javanese and Tagalog as the smallest language pairs in the dataset. We perform our experiments on multiple dataset scales: 80K (small), 150K (medium), and 500K (large). Our primary goal is to test the method; therefore, we do not perform experiments on the whole dataset, leaving this to future work. For additional experiments, we utilize the Hungarian dataset from Small Track #1. All the dataset statistics are provided in Table 4 for the initial dataset and for the datasets created for our experiments.

Finno-Samic Languages Dataset. Regarding the dataset compiled from the Finno-Ugric benchmark (Yankovskaya et al., 2023), we stick to the Sami languages as the only option matching our criteria: parallel training datasets of different sizes with the same output language (Finnish) for those pairs, parallel development and test datasets of good quality. Unfortunately, such data is available only for Finno-Samic languages[3], such as tartuNLP/finno-ugric-benchmark North Sami, South Sami, Inari Sami, Skolt Sami. The dataset statistics are presented in Table 5. In future experiments, we plan to extend the datasets to other languages and directions from the benchmark.

4.2 Baselines

For our baselines, we consider fine-tuning to the target language both with and without various forms of prior continual pretraining:

- FT_{All} — Fine-tuning to all languages including the target language;
- FT_{NoT} — Fine-tuning to all languages except the target language;
- FT_{OnlyT} — Fine-tune to the target language only;
- CP_{All} — Continuous Pretraining to all languages, followed by additional fine-tuning to the target language;
- CP_{NoT} — Continuous Pretraining to all languages but the target, followed by additional fine-tuning to the target language.

[3] https://huggingface.co/datasets/tartuNLP/finno-ugric-benchmark, https://huggingface.co/datasets/tartuNLP/finno-ugric-train

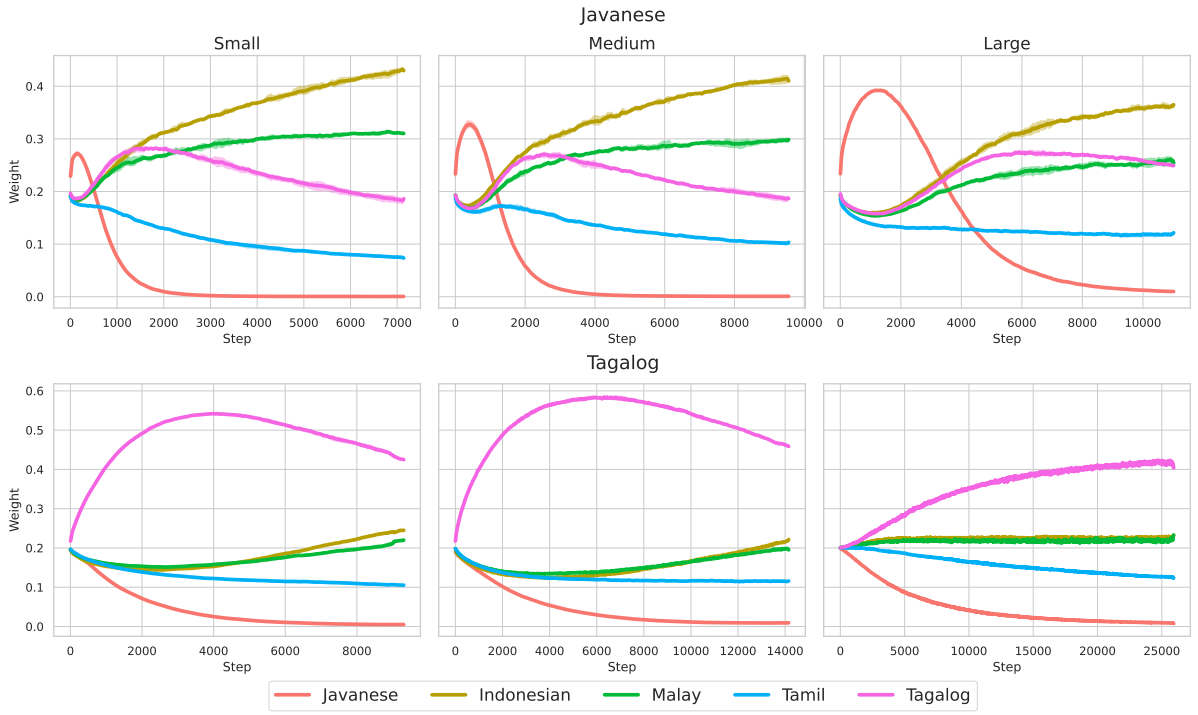

Figure 1: Weights distribution for South East Asian languages. Target languages and data sizes are in captions.

We use the M2M100 model with 418M parameters as our base model (Fan et al., 2020). For Finno-Ugric languages, special language tokens are added and learned since the model was not pretrained for those languages. More training details and configurations are provided in Appendix A.

4.3 Evaluation

We use SpBLEU metrics in our evaluation as in Sutawika and Cruz (2021), utilizing SacreBLEU (Post, 2018). The generation parameters are adopted from Xie et al. (2021), employing beam search with 4 beams, and the temperature set to 1.

5 Results and Discussion

Tables 1 and 2 show that `MeritFed` is indeed helpful during training: our approach achieves better performance for most setups and languages. for Javanese and Tagalog languages (small and medium) and for Sami languages of comparable sizes (South, Scolt, and Inari). We report scores for approaches without Continuous Pretraining in Appendix B, as they are always worse than with CP.

Impact of Aggregation Weights. We can see that the methods assign higher weights to the target language at first, followed by a drop, while other weights increase. Therefore, Javanese benefits more from the Indonesian language, while Tagalog's, higher-weighted languages are Indonesian and Malay. Interestingly, while spoken in South East Asia, the Tamil language does not belong to the same language family as the others. This fact is reflected in Figure 1: Tamil always contributes less than other languages. For Sami languages, North Sami seems always to be the most beneficial.

No Overfitting. An important observation is that the algorithm helps to prevent the model from overfitting: the weight of the target language decreases once the model learns the small amount of data available for the target language; additional languages serve as regularization to keep the model converging. Probably, that partially explains the non-zero weights of Tamil, which does not belong to the same language family, although being spoken in South East Asia.

Unrelated Language. To check the hypothesis that unrelated language serves as regularization, we conducted an additional experiment and added the Hungarian language from the Finno-Ugric family to training. As shown in Figure 3, its weights are also non-zero. Moreover, the SpBLEU scores remained nearly consistent across all MD parameters and Adaptive Batch configurations, supporting the regularization role of additional languages.

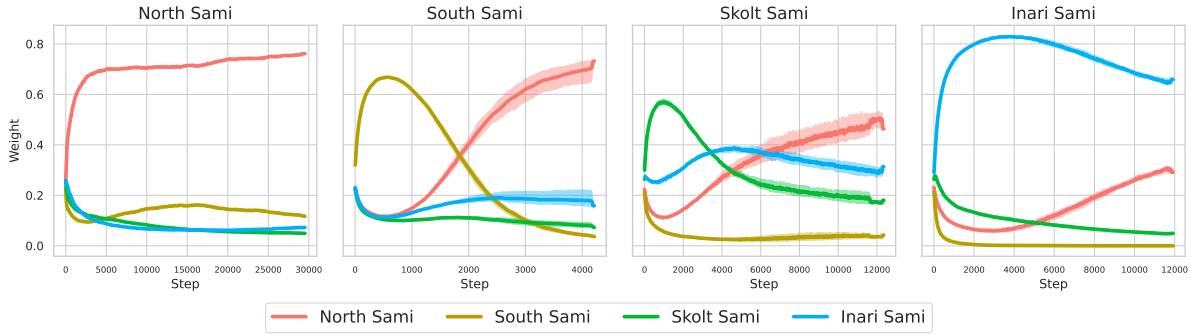

Figure 2: Weights distribution across Finno-Samic languages. Target languages are mentioned in captions.

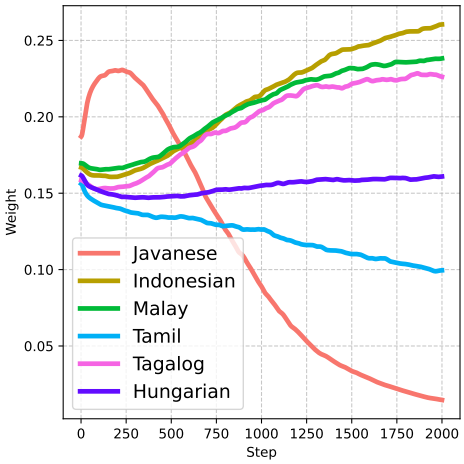

Figure 3: Weights distribution for target Indonesian language with unrelated Hungarian included.

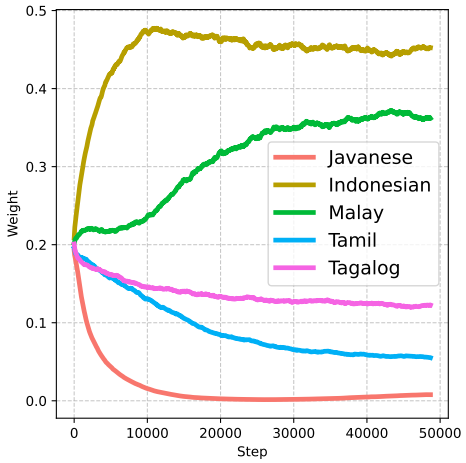

Figure 4: Weights distribution for languages with target Indonesian on *small* subset.

Size of the Target Language Dataset. For Tagalog-large and North Sami, the algorithm relies on the target language dataset more than on additional languages and does not outperform the Continuous Pretraining baseline. On the contrary, for small and medium datasets, the algorithm needs from 2 to 10 times fewer main gradient steps to outperform the baselines.

We assume that this happens because the amount of data from the target language is enough, and the algorithm keeps assigning high weights to the target language and trains the model on the target language only. Another possible reason for the inferior performance could be excessive gradient steps involving non-target languages. This might "distract" the model and fail to provide significant benefits. Since at each step, we compute the stochastic gradients for other languages, too, the method does not pass the whole dataset of the target language, given the computational resources for the experiment. Therefore, the method does not utilize all potentially useful information from the target language.

This hypothesis is supported by an additional experiment on the Indonesian language as the target language with the biggest dataset to see the distribution of the weights for a longer number of steps ($\sim$ 50K). From Figure 4, we observe the evolution of corresponding aggregation weights: it keeps growing during the training, which indicates its significantly higher importance on the model quality than other languages. Once the model learns the dataset better, the weight of the Indonesian slightly decreases and gets stuck, while the weight of the Malay starts to grow. We assume that these observations might be useful for further experiments: languages that stop contributing to the algorithm convergence can be "dropped" during training.

Adaptive Batch Experiments. We hypothesize that leveraging high-resource languages could improve gradient approximation by providing more samples. Based on this, we develop an Adaptive Batch procedure. This method allocates the total batch size (512 in our experiments) and samples batch size from the total size for each language proportionally to the percentage of each language

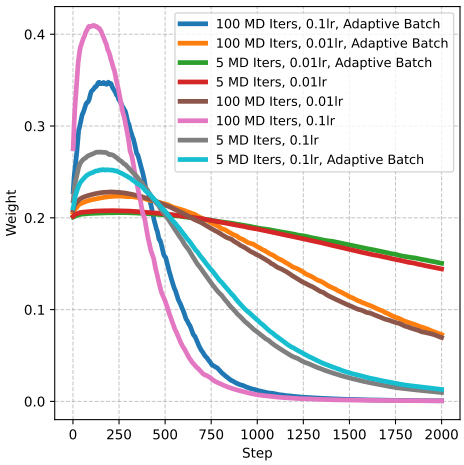

Figure 5: Weights for target language (Javanese-small) with different Mirror Descent parameters.

present in the dataset. Thus, high-resource languages receive larger batch sizes. To optimize convergence, we set batch size limits, with a lower bound of 32 and an upper bound of 128, as shown to be effective in previous studies (Keskar et al., 2017; Bengio, 2012).

However, our results indicate that the Adaptive Batch procedure is rarely beneficial. We believe this is due to the downside of better gradient approximation. Our method suggests that assigning higher weight to high-resource languages due to their well-estimated gradients may hinder the learning of the target language. This is illustrated in Figure 5, where adding an Adaptive Batch leads to a lower weight for the target language.

Mirror Descent Parameters Impact. We conduct experiments with various `MeritFed` settings, adjusting the Mirror Descent learning rate to 0.1 and 0.01 and the number of iterations to 5 or 100. These experiments are performed on the *small* subset of the South East Asian dataset, using Javanese as the target language.

As shown in Table 3, the Mirror Descent parameters have little impact, with no clear trend emerging[4]. For 5 iterations, a higher learning rate, and no Adaptive Batch, the algorithm performs better when no unrelated language is present. However, this changes when an unrelated language is included. For 100 iterations, a lower learning rate, and no Adaptive Batch, the model consistently yields better results. Based on those observations, we have chosen 5 MD iterations with a learning rate of 0.1 for all experiments due to its high performance and faster computation times.

[4]We conjecture that (Stochastic) Mirror Descent struggles to solve the auxiliary problem from Line 6 with sufficiently good accuracy since it can be viewed as a variant of `SGD` for problems with non-Euclidean prox-structure and `SGD` is known to perform poorly in NLP tasks (Zhang et al., 2020). Investigation of other alternatives (e.g., MD version of `Adam`) is a prominent direction for future research.

MD Iterations	Learning Rate	Adaptive Batch	SpBLEU	
			Relevant	+Irrelevant
5	0.1	-	19.735	19.786
		+	19.674	19.620
	0.01	-	19.724	19.724
		+	19.671	19.810
100	0.1	-	19.702	19.645
		+	19.568	19.591
	0.01	-	19.754	19.719
		+	19.739	19.639

Table 3: Scores and Settings Grouped by Mirror Descent iterations for the Javanese-small dataset.

Theoretical Results. We prove that under certain assumptions on the underlying optimizer `OptStep`, `MeritFed` converges to the neighborhood of the solution of the target problem when (i) the learning rate is small enough, (ii) $\widehat{\boldsymbol{D}}$ is sufficiently large such that $f_{\widehat{\boldsymbol{D}}}$ is close to $f_{\mathcal{D}}$, and (iii) the auxiliary problem in Line 6 is solved with a good accuracy. In Appendix D, we provide missing theoretical details (including the proofs) and show that `SGD`, `RMSProp`, `AdaGrad-Norm` satisfy our assumptions.

6 Conclusion

In this paper, we implement the `MeritFed` algorithm from the Personalised Federated Learning to the Low-Resource Machine Translation task. We show that it can achieve better results than traditional approaches and requires 2 to 10 times fewer gradient steps than baselines (e.g., 8K vs. 85K, 12K vs. 23K). `MeritFed` also allows us to observe the weight distribution between the target and related languages: Javanese benefits more from the Indonesian language, while for Tagalog, the most important languages are Indonesian and Malay. Different weights for different languages also prevent the model from overfitting: after learning the target language dataset, its weights are dropped down while other weights start growing. Another takeaway is about the target dataset size: the bigger the dataset is, the more the algorithm keeps relying on it rather than on the auxiliary languages. This might result in worse model performance and "distract" the model from convergence.

Limitations

- We report results only on Low-Resource MT, while a wide variety of NLP tasks are available. We leave further investigation of `MeritFed` to other NLP tasks for future work.
- We report results only on M2M100, while numerous LLMs are available. An alternative model with the `MeritFed` algorithm applied could further improve the results. The research focuses on the algorithm application to the LRMT task and not on an exhaustive search of all LLM models.
- We limit our dataset in terms of size and language variety because of high computational costs and limited resources available.
- Our setup with the limited amount of languages and training data used is not designed to directly compare with the existing approaches.
- We retain all languages during training, even those that do not contribute, which affects the efficiency of the training procedure.

Ethical Statement

In our research, we utilize the M2M100 model, which has been pre-trained on a diverse MT corpus, including user-generated content. The datasets we use for additional model training have already been presented in WMT-21 Shared Task and Finno-Ugric Benchmark. Although we expect them to be filtered from harmful content, it is important to recognize that some biases may still persist in the model outputs.

This acknowledgment does not undermine the validity of our methods. We have designed our techniques to be flexible, allowing them to be applied to alternative pre-trained models that have undergone more rigorous debiasing processes. To the best of our knowledge, aside from the challenge of mitigating inherent biases, our work does not raise any additional ethical concerns.

References

Yoshua Bengio. 2012. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade: Second edition*, pages 437–478. Springer.

Sari Dewi Budiwati, Tirana Fatyanosa, Mahendra Data, Dedy Rahman Wijaya, Patrick Adolf Telnoni, Arie Ardiyanti Suryani, Agus Pratondo, and Masayoshi Aritsugi. 2021. To optimize, or not to optimize, that is the question: TelU-KU models for WMT21 large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 387–397, Online. Association for Computational Linguistics.

Hong Chen, Xin Wang, Chaoyu Guan, Yue Liu, and Wenwu Zhu. 2022. Auxiliary learning with joint task and data scheduling. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 3634–3647. PMLR.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR.

Severino Da Dalt, Joan Llop, Irene Baucells, Marc Pamies, Yishi Xu, Aitor Gonzalez-Agirre, and Marta Villegas. 2024. FLOR: On the effectiveness of language adaptation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7377–7388, Torino, Italia. ELRA and ICCL.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized Federated Learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond English-centric multilingual machine translation. *Journal of Machine Learning Research*, 22:1–48.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. 2020. Efficient neural machine translation for low-resource languages via exploiting related languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 162–168, Online. Association for Computational Linguistics.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of Low-Resource Machine Translation. *Computational Linguistics*, 48(3):673–732.

Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. 2020. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 33:2304–2315.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2.

Wenshuai Huo, Xiaocheng Feng, Yichong Huang, Chengpeng Fu, Hui Wang, and Bing Qin. 2024. Gradient consistency-based parameter allocation for multilingual neural machine translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7901–7912, Torino, Italia. ELRA and ICCL.

Ayyoob ImaniGooghari, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2017. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, San Diego, CA, USA.

Jakub Konecnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 8.

Panteleimon Krasadakis, Evangelos Sakkopoulos, and Vassilios S. Verykios. 2024. A survey on challenges and advances in natural language processing with a focus on legal informatics and low-resource languages. *Electronics*, 13(3).

Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. 2020. Survey of personalization techniques for federated learning. In *2020 fourth world conference on smart trends in systems, security and sustainability (WorldS4)*, pages 794–797. IEEE.

Baohao Liao, Shahram Khadivi, and Sanjika Hewavitharana. 2021. Back-translation for large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 418–424, Online. Association for Computational Linguistics.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André F. T. Martins, and Hinrich Schütze. 2024. MaLA-500: Massive language adaptation of large language models. *CoRR*, abs/2401.13303.

Varvara Logacheva, Denis Teslenko, Artem Shelmanov, Steffen Remus, Dmitry Ustalov, Andrey Kutuzov, Ekaterina Artemova, Chris Biemann, Simone Paolo Ponzetto, and Alexander Panchenko. 2020. Word sense disambiguation for 158 languages using word embeddings only. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5943–5952, Marseille, France. European Language Resources Association.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *CoRR*, abs/2106.13736.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Alice Millour, Lorenza Brasile, Alberto Ghia, and Laurent Kevers. 2024. Agettivu, aggitivu o aghjettivu? POS tagging Corsican dialects. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 600–608, Torino, Italia. ELRA and ICCL.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. 1983. Problem complexity and method efficiency in optimization.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2023. Neural machine translation for low-resource languages: A survey. *ACM Comput. Surv.*, 55(11):229:1–229:37.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Fynn Schröder and Chris Biemann. 2020. Estimating the influence of auxiliary tasks for multi-task learning of sequence tagging tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2971–2985, Online. Association for Computational Linguistics.

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. 2009. Stochastic convex optimization. In *COLT*, volume 2, page 5.

Matthew Streeter and H. Brendan McMahan. 2010. Less regret via online conditioning. *arXiv preprint arXiv:1002.4862*.

Lintang Sutawika and Jan Christian Blaise Cruz. 2021. Data processing matters: SRPH-konvergen AI's machine translation system for WMT'21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 431–438, Online. Association for Computational Linguistics.

Maali Tars, Andre Tättar, and Mark Fishel. 2022. Cross-lingual transfer from large multilingual translation models to unseen under-resourced languages. *Balt. J. Mod. Comput.*, 10(3).

Nazarii Tupitsa, Samuel Horváth, Martin Takáč, and Eduard Gorbunov. 2024. Federated learning can find friends that are beneficial. *arXiv preprint arXiv:2402.05050*.

Rui Wang, Xu Tan, Renqian Luo, Tao Qin, and Tie-Yan Liu. 2021. A survey on low-resource neural machine translation. In *IJCAI*, pages 4636–4643. ijcai.org.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. 2019. AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6677–6686. PMLR.

Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush, and Francisco Guzmán. 2021. Findings of the WMT 2021 shared task on large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 89–99, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hongda Wu and Ping Wang. 2021. Fast-convergent federated learning with adaptive weighting. *IEEE Transactions on Cognitive Communications and Networking*, 7(4):1078–1088.

Wanying Xie, Bojie Hu, Han Yang, Dong Yu, and Qi Ju. 2021. TenTrans large-scale multilingual machine translation system for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 439–445, Online. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual machine translation systems from Microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 446–455, Online. Association for Computational Linguistics.

Lisa Yankovskaya, Maali Tars, Andre Tättar, and Mark Fishel. 2023. Machine translation for low-resource Finno-Ugric languages. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 762–771, Tórshavn, Faroe Islands. University of Tartu Library.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. 2018. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. 2020. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393.

A Dataset and Model Details

In addition to dataset scaling, we also add a preprocessing step: from a deeper look into the data, we can see that some translations contain code snippets, HTML, and pairs containing different addresses and numbers in the input language and output language. To avoid such data, we filter the sentences in the training set so that (i) input and output length in tokens is not less than 5 tokens and not larger than 256 tokens, as only a negligible portion of the data exceeded this limit; (ii) we keep sentences with the numbers matching in both input and output; (iii) we keep alphanumeric sentences with basic punctuation only. We also check that both datasets we apply do not contain personally identifying information or offensive content.

We used M2M100 as a base model (Fan et al., 2021), MIT Licensed. In CP setting, we pretrained all models on all languages for a maximum of 10 epochs, with the best-performing checkpoint selected for later fine-tuning. Fine-tuning was conducted for up to 60 epochs, and the best-performing checkpoint was reported. The `MeritFed` model was trained until the score stopped improving, with a maximum computation time of four days.

Training parameters included a fixed batch size of 64 and a learning rate of 3e-5. We used a Cosine Annealing Scheduler with a minimum learning rate of 1e-5. The baseline optimizer was `Adam`, with $\beta_1 = 0.9$ and $\beta_2 = 0.98$, in line with previous studies (Xie et al., 2021).

Our implementation primarily relied on PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020) libraries. All our artifacts are licensed under Apache 2.0.

Input Language	Total	Filtered Train			Val	Test
		Small	Medium	Large		
Indonesian	54M	37K	74K	259K	1K	1K
Malay	13M	26K	53K	185K	1K	1K
Tagalog	2M	10K	20K	70K	1K	1K
Tamil	13M	5K	10K	35K	1K	1K
Javanese	3M	776	1.5K	5K	1K	1K

Table 4: Dataset statistics for South East Asian languages. Total denotes the original dataset size in sequences, Filtered small, medium and large train are the subsets used for experiments.

Input Language	Train	Val	Test
North Sami	61,559	200	500
Inari Sami	8,750	200	500
Skolt Sami	1,998	200	500
South Sami	1,734	200	500

Table 5: Dataset statistics for Finno-Samic languages.

B Extended Baseline Results

In this section, we report the results with more baseline settings:

Method	Tagalog						Java					
	79K		155K		555K		79K		128K		555K	
	Score	Steps	Score	Steps	Score	Steps	Score	Steps	Score	Steps	Score	Steps
CP_{All}	29.24 ± 0.06	21K	30.99 ± 0.04	40K	33.89 ± 0.15	124K	19.43 ± 0.14	12K	20.05 ± 0.12	25K	20.97 ± 0.13	87K
CP_{NoT}	28.72 ± 0.16	15K	30.50 ± 0.12	42K	33.74 ± 0.19	129K	19.46 ± 0.12	12K	19.95 ± 0.12	25K	21.19 ± 0.09	89K
FT_{OnlyT}	28.69 ± 0.10	8K	30.48 ± 0.05	44K	33.88 ± 0.07	52K	19.23 ± 0.02	500	19.69 ± 0.01	1K	20.75 ± 0.10	3.5K
FT_{All}	24.78 ± 0.02	12K	26.53 ± 0.17	25K	30.02 ± 0.03	79K	19.26 ± 0.02	12K	19.28 ± 0.07	25K	19.92 ± 0.06	85K
FT_{NoT}	20.45 ± 0.08	7K	20.41 ± 0.07	11K	20.34 ± 0.09	53K	18.73 ± 0.02	12K	18.80 ± 0.04	25K	18.94 ± 0.09	85K
MeritFed	29.73 ± 0.04	14K	31.42 ± 0.07	14K	33.53 ± 0.27	47K	19.74 ± 0.03	2K	20.23 ± 0.11	3K	21.44 ± 0.13	8K

Table 6: Mean SpBLEU scores and the number of steps required to reach them for all baselines and MeritFed within the different data sizes of Javanese and Tagalog languages

Method	SMA		SMS		SMN		SME	
	Score	Steps	Score	Steps	Score	Steps	Score	Steps
CP_{All}	11.60 ± 0.29	23K	44.90 ± 0.12	25K	51.39 ± 0.05	30K	39.78 ± 0.08	69K
CP_{NoT}	11.09 ± 0.24	23K	43.40 ± 0.13	25K	50.14 ± 0.04	31K	39.30 ± 0.18	65K
FT_{OnlyT}	9.44 ± 0.20	1.5K	38.83 ± 0.31	2K	48.70 ± 0.14	8K	39.26 ± 0.33	53K
FT_{All}	5.56 ± 0.29	21K	34.11 ± 0.23	23K	44.62 ± 0.10	23K	33.57 ± 2.34	12K
FT_{NoT}	2.38 ± 0.09	16K	11.62 ± 0.37	23K	16.63 ± 0.35	16K	10.16 ± 0.16	2K
MeritFed	13.26 ± 0.17	2.5K	50.27 ± 0.17	12K	52.08 ± 0.01	12K	38.526 ± 1.39	30K

Table 7: Mean SpBLEU scores and the number of steps required to reach them for all baselines and MeritFed within Finno-Samic low-resource languages

C Illustrative Experiment with Mean Estimation Problem

In this section, we provide an illustrative experiment with the mean estimation problem. That is, the goal is to solve the following minimization problem:

$$\min_{x \in \mathbb{R}^d} \left\{ f_{\mathcal{D}}(x) := \mathbb{E}_{\xi \sim \mathcal{D}}[\|x - \xi\|^2] \right\},$$

where $\mathcal{D} := \mathcal{N}(0, \boldsymbol{I})$ is a standard Gaussian distribution. One can show that the optimal value equals $\mathbb{E}_{\xi \sim \mathcal{D}}\|\xi\|^2 = d$, which is attained at $x^* = 0$. Next, we consider three datasets: $\boldsymbol{D}_1$ is sampled from the target distribution $\mathcal{D}$, $\boldsymbol{D}_2$ is sampled from close distribution $\mathcal{N}(\mu \mathbf{1}, \boldsymbol{I})$, where $\mu = 0.0001$ and $\mathbf{1} := (1, \ldots, 1)^\top \mathbb{R}^d$, and $\boldsymbol{D}_3$ is sampled from quite different distribution $\mathcal{N}(e, \boldsymbol{I})$, where e is some randomly precomputed unit vector. The sizes of the input dataset are: $|\boldsymbol{D}_1| = 20$, $|\boldsymbol{D}_2| = 1000$, $|\boldsymbol{D}_3| = 1000$. Therefore, this situation resembles training for the low-resource language, when two high-resource languages are available. We take mini-batch of 10% for each dataset to compute $g_i(x^t)$ in `MeritFed` and use simple `SGD` as `OptStep`. Target validation dataset $\widehat{\boldsymbol{D}}$ is sampled from $\mathcal{D}$ (same distribution as for $\boldsymbol{D}_1$) and has size $|\widehat{\boldsymbol{D}}| = 100$ (though only mini-batch of 10 samples from $\widehat{\boldsymbol{D}}$ is used at each iteration to perform a computation of aggregation weight w^{t+1}). To solve the problem in Line 6, we run `MD` with learning rate 10.

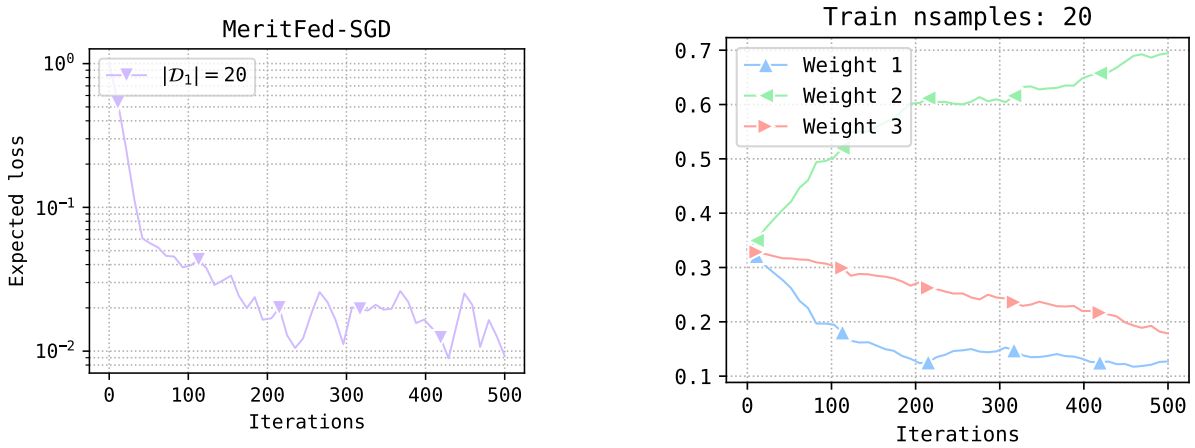

Figure 6: Mean Estimation: $\mu = 0.0001$, MD learning rate = 10.

The results are presented in Figure 6. We see that the weight for the first and the third datasets decrease during the training, while the weight for the second dataset increases and remains the largest one. Such a behavior is natural since the batchsize for the target dataset is much smaller than for the second dataset (2 and 100 respectively) and since the second dataset comes from very close distribution to the target one it is more beneficial to use slightly biased but less noisy updates from the second dataset than unbiased but noisy updates from the first dataset. As for the third dataset, its weight decreases since it comes from completely different distribution.

Overall, the result of this experiment are quite consistent with the ones we obtained for Javanese language where the weight for the target language also becomes the smallest after certain number of steps and the highest weight is assigned to close but different language (Indonesian), see Figure 1.

D Theoretical Results: Complete Statements and Proofs

D.1 Preliminaries

In this section, we provide the details on the theoretical convergence results for `MeritFed`. For notational convenience, we assume that $\boldsymbol{D}_i$ comes from distribution $\mathcal{D}_i$ and denote the corresponding expected loss function as f_i for all $i = 1, \ldots, n$. Therefore, according to the introduced notation f_1 and $f_{\mathcal{D}}$ denote the same loss function. Similarly to the setup considered by Tupitsa et al. (2024), we denote the set of indices such that $\mathcal{D}_i = \mathcal{D}_1$: $\mathcal{G} := \{i \in \{1, \ldots, n\} \mid \mathcal{D}_i = \mathcal{D}_1\}$. In other words, for every $i \in \mathcal{G}$ dataset $\boldsymbol{D}_i$ comes from the target distribution and, thus, should be beneficial for the training.

Next, we make the following standard assumption about the stochastic gradients.

Assumption 1. *For all $i \in \mathcal{G}$ the stochastic gradient $g_i(x)$ is an unbiased estimator of $\nabla f_i(x)$ with bounded variance, i.e., $\mathbb{E}_{\xi_i \sim \mathcal{D}_i}[g_i(x)] = \nabla f_i(x)$ and for some $\sigma \geq 0$*

$$\mathbb{E}_{\xi_i \sim \mathcal{D}_i}\left[\|g_i(x) - \nabla f_i(x)\|^2\right] \leq \sigma^2. \tag{1}$$

Let w^{ideal} denote a weight vector containing equal non-zero weights only for the datasets from the target distribution. If Assumption 1 holds, then due to the independence of $\{g_i(x)\}_{i \in \mathcal{G}}$

$$\mathbb{E}_{\xi_i}\left[\left\|\sum_{i=1}^n w_i^{\text{ideal}} g_i(x) - \nabla f_1(x)\right\|^2\right] = \mathbb{E}_{\xi_i}\left[\left\|\frac{1}{|\mathcal{G}|}\sum_{i \in \mathcal{G}} g_i(x) - \nabla f_1(x)\right\|^2\right] \leq \frac{\sigma^2}{|\mathcal{G}|} \equiv \sigma_*^2. \tag{2}$$

We also assume that the objective is L-smooth

Assumption 2. *f_1 is L-smooth, i.e., $\forall\, x, y \in \mathbb{R}^d$*

$$f_1(x) \leq f_1(y) + \langle \nabla f_1(y), x - y\rangle + \frac{L}{2}\|x - y\|^2. \tag{3}$$

For the sake of brevity, we will also use the following notation:

$$x^{t+1}(w) = \texttt{OptStep}\left(x^t, \sum_{i=1}^n w_i g_i(x^t), \gamma_t\right).$$

D.2 Generic Scheme of the Proof

The proof for `MeritFed-SGD` from (Tupitsa et al., 2024) is based on the assumption that the auxiliary problem can be solved with δ error:

$$\mathbb{E}\left[f_1(x^{t+1}) | x^t, \xi^t\right] - \min_{w \in \Delta_1^n} f_1\left(x^{t+1}(w)\right) \leq \delta, \tag{4}$$

and the following inequality

$$\min_{w \in \Delta_1^n} f_1\left(x^{t+1}(w)\right) \leq f_1(x^{t+1}(w^{\text{ideal}})), \tag{5}$$

which holds by the definition of the minimum. These two inequalities together imply

$$\mathbb{E}\left[f_1(x^{t+1}) | x^t\right] \leq \mathbb{E}\left[f_1(x^{t+1}(w^{\text{ideal}})) | x^t\right] + \delta. \tag{6}$$

The rest of the proof for `MeritFed-SGD` follows the same scheme as for `SGD` that uses $\sum_{i=1}^n w_i^{\text{ideal}} g_i(x)$ as the stochastic gradient, i.e., as for the method $x^{t+1} = x^t - \gamma \sum_{i=1}^n w_i^{\text{ideal}} g_i(x^t) = x^t - \frac{\gamma}{|\mathcal{G}|}\sum_{i \in \mathcal{G}} g_i(x^t)$.

We noticed, that convergence result of `MeritFed` envelope can be obtained in the case when the analysis of the method being enveloped uses only two subsequent points and relies on the analysis of the inequality $\mathbb{E}[f_1(x^{t+1})] \leq \mathbb{E}[f_1(x^t)] + \Delta_t$, where Δ_t is some additional iteration-dependent term. Then, using (6), one can show that `MeritFed` version of the method decreases expected function value not less then the ideal update at each iteration (up to the error term of solving the problem in Line 6). In the next two subsections, we provide the results for `MeritFed-RMSProp` and `MeritFed-AdaGrad-Norm`.

D.3 Special Case: RMSProp

In this subsection, we consider RMSProp as OptStep, i.e.,

$$\texttt{OptStep}(x^t, g^t, \gamma_t) = x^t - \frac{\gamma_t}{b_t} g^t, \quad b_t = \sqrt{\beta_2 b_{t-1}^2 + (1-\beta_2)(g^t)^2} + \epsilon,$$

where all arithmetical operations (multiplication, division, summation, taking the square/square root) are coordinate-wise. We emphasize that RMSProp can be seen as Adam without momentum ($\beta_1 = 0$).

We base our proof on the one from (Zaheer et al., 2018), that additionally uses the following assumption.

Assumption 3. *Each component of the stochastic gradient $g_i(x)$ for $i \in \mathcal{G}$ is bounded, i.e.,*

$$\left\| [g_i(x)]_j \right\| \leq G. \tag{7}$$

Theorem 1. *Let Assumptions 1, 2, 3 hold. If Line 6 is solved with error $\delta \geq 0$ (see (4)), then* MeritFed-RMSProp *with $\gamma_t = \gamma \leq \frac{\epsilon}{2L}$ and $\beta_2 \geq 1 - \frac{\epsilon^2}{16G^2}$ after T iterations satisfy*

$$\min_{t=0,\ldots,T-1} \mathbb{E}\|\nabla f_1(x^t)\|^2 \leq 2(\sqrt{\beta_2}G + \epsilon) \times \left[\frac{\left(f_1(x^0) - f_1(x^*)\right)}{\gamma T} + \sigma_*^2 \left(\frac{\gamma G \sqrt{1-\beta_2}}{\epsilon^2} + \frac{L\gamma^2}{2\epsilon^2} \right) + \frac{\delta}{\gamma} \right].$$

Proof. We start with the following inequality from the page 13 of (Zaheer et al., 2018)

$$\mathbb{E}[f_1(x^{t+1}(w^{\text{ideal}}))|x^t] \leq f_1(x^t) - \frac{\gamma_t}{2\left(\sqrt{\beta_2}G + \epsilon\right)} \left\|\nabla f_1(x^t)\right\|^2 + \left(\frac{\gamma_t G \sqrt{1-\beta_2}}{\epsilon^2} + \frac{L\gamma_t^2}{2\epsilon^2} \right) \sigma_*^2$$

in a slightly adjusted form. In fact, this inequality holds for any x^t and ideally aggregated gradients $\sum\limits_{i=1}^{n} w_i^{\text{ideal}} g_i(x^t)$. Applying (6), we get

$$\mathbb{E}\left[f_1(x^{t+1})|x^t\right] \leq f_1(x^t) - \frac{\gamma_t}{2\left(\sqrt{\beta_2}G + \epsilon\right)} \left\|\nabla f_1(x^t)\right\|^2 + \left(\frac{\gamma_t G \sqrt{1-\beta_2}}{\epsilon^2} + \frac{L\gamma_t^2}{2\epsilon^2} \right) \sigma_*^2 + \delta.$$

Following the same steps of the rest of the proof from (Zaheer et al., 2018), we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla f_1\left(x^t\right)\right\|^2 \leq 2(\sqrt{\beta_2}G + \epsilon) \times \left[\frac{\left(f_1(x^0) - f_1(x^*)\right)}{\gamma T} + \sigma_*^2 \left(\frac{\gamma G \sqrt{1-\beta_2}}{\epsilon^2} + \frac{L\gamma^2}{2\epsilon^2} \right) + \frac{\delta}{\gamma} \right],$$

where $\gamma_t = \gamma \leq \frac{\epsilon}{2L}$ is used. It remains to notice that $\min\limits_{t=0,\ldots,T-1} \mathbb{E}\|\nabla f_1(x^t)\|^2 \leq \frac{1}{T}\sum\limits_{t=0}^{T-1} \mathbb{E}\left\|\nabla f_1\left(x^t\right)\right\|^2$. □

D.4 Special Case: AdaGrad-Norm

In this subsection, we consider AdaGrad-Norm (Ward et al., 2019) as OptStep, i.e.,

$$\texttt{OptStep}(x^t, g^t, \gamma_t) = x^t - \frac{\gamma_t}{b_{t+1}} g^t, \quad b_{t+1} = \sqrt{b_t^2 + \|g^t\|^2}.$$

We base our proof on the one from (Ward et al., 2019), that additionally uses the following assumption.

Assumption 4. *Gradients $\nabla f_i(x)$ are uniformly bounded for $i \in \mathcal{G}$:*

$$\|\nabla f_i(x)\| \leq G. \tag{8}$$

Theorem 2. *Let Assumptions 1, 2, 4 hold. If Line 6 is solved with error $\delta \geq 0$ (see (4)), then* MeritFed-AdaGrad-Norm *with after T iterations satisfy*

$$\min_{t \leq T} \left(\mathbb{E}\left[\|\nabla f_1(x^t)\|^{\frac{4}{3}} \right] \right)^{\frac{3}{2}} \leq \left(\frac{2b_0}{T} + \frac{4(G + \sigma_*)}{\sqrt{T}} \right) C_F,$$

where

$$C_F = \frac{\delta T}{\gamma} + \frac{f_1(x^0) - f_1(x^*)}{\gamma} + \frac{4\sigma_* + \gamma L}{2} \log\left(\frac{20T\left(\sigma_*^2 + G^2\right)}{b_0^2} + 10 \right).$$

Proof. Notating $\widetilde{g}^t = \frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}} g_i(x^t)$ and $\widetilde{b}_{t+1} = \sqrt{b_t^2 + \|\widetilde{g}^t\|^2}$ we rewrite the first line of the main proof from (Ward et al., 2019) as

$$\begin{aligned}\frac{f_1(x^{t+1}(w^{\text{ideal}})) - f_1(x^t)}{\gamma} &\leq \frac{-\langle \nabla f_1(x^t), \widetilde{g}^t\rangle}{\widetilde{b}_{t+1}} + \frac{\gamma L}{2\widetilde{b}_{t+1}^2}\|\widetilde{g}^t\|^2 \\ &= -\frac{\|\nabla f_1(x^t)\|^2}{\widetilde{b}_{t+1}} + \frac{\langle \nabla f_1(x^t), \nabla f_1(x^t) - \widetilde{g}^t\rangle}{\widetilde{b}_{t+1}} + \frac{\gamma L\|\widetilde{g}^t\|^2}{2\widetilde{b}_{t+1}^2}.\end{aligned}$$

Applying (4) and (5), we get the following inequality:

$$\frac{f_1(x^{t+1}) - \delta - f_1(x^t)}{\gamma} \leq -\frac{\|\nabla f_1(x^t)\|^2}{b_{t+1}} + \frac{\langle \nabla f_1(x^t), \nabla f_1(x^t) - g^t\rangle}{b_{t+1}} + \frac{\gamma L\|g^t\|^2}{2b_{t+1}^2}.$$

Then, following the same steps as in the main proof from (Ward et al., 2019), we derive

$$\min_{t\leq T}\left(\mathbb{E}\left[\|\nabla f_1(x^t)\|^{\frac{4}{3}}\right]\right)^{\frac{3}{2}} \leq \left(\frac{2b_0}{T} + \frac{4(G+\sigma_*)}{\sqrt{T}}\right)C_F,$$

where

$$C_F = \frac{\delta T}{\gamma} + \frac{f_1(x^0) - f_1(x^*)}{\gamma} + \frac{4\sigma_* + \gamma L}{2}\log\left(\frac{20T\left(\sigma_*^2 + G^2\right)}{b_0^2} + 10\right).$$

This finishes the proof. □