Pretrained Models for Multilingual Federated Learning

Orion Weller*, Marc Marone*, Vladimir Braverman, Dawn Lawrie, Benjamin Van Durme
Johns Hopkins University
oweller@cs.jhu.edu,mmarone1@jhu.edu

Abstract

Since the advent of Federated Learning (FL), research has applied these methods to natural language processing (NLP) tasks. Despite a plethora of papers in FL for NLP, no previous works have studied how multilingual text impacts FL algorithms. Furthermore, multilingual text provides an interesting avenue to examine the impact of non-IID text (e.g. different languages) on FL in naturally occurring data. We explore two multilingual language tasks, language modeling and machine translation, using differing federated settings as well as non-federated learning algorithms. Our results show that using pretrained models reduces the negative effects of FL, helping them to perform near or better than centralized (no privacy) learning, even when using non-IID partitioning.[1]

1 Introduction

Federated learning (FL) is a machine learning technique that trains an algorithm across multiple distributed clients holding local data samples, without ever storing client data in a central location (Konečnỳ et al., 2016; McMahan et al., 2017). These techniques are appealing for those who wish to learn from data in a privacy-preserving way, without ever transmitting the data off of a client device. FL becomes essential when data is especially sensitive, as is the case at hospitals, legal firms, financial institutions, or in countries that enact legislation concerning data privacy (such as the EU's GDPR or the US's HIPAA).

FL has been applied to problems in natural language processing (NLP) since its inception, particularly in use of the language modeling task (Yang et al., 2018; Hard et al., 2018; Ramaswamy et al., 2019; Chen et al., 2019a; Ji et al., 2019; Stremmel and Singh, 2020). Another large area of FL research is focused on improving performance when the data is non identically independently distributed (non-IID). In these cases, many works have shown that FL performance is sub-par with respect to centralized learning methods (Konečnỳ et al., 2016; Hard et al., 2018; Lin et al., 2021).

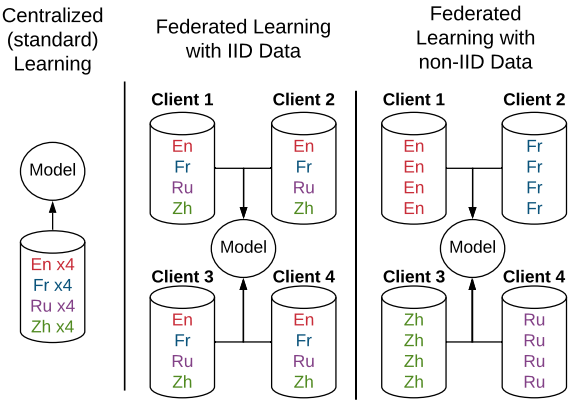

Figure 1: A depiction of different learning strategies with Federated Learning (FL) and multilingual data, with 4 clients and 16 instances from En, Fr, Ru, and Zh in this toy example. Black lines indicate gradient flow. Centralized learning is the standard training method (no privacy), FL with IID data partitions the data into IID data subsets for each client, while FL with non-IID data has the languages separated across clients.

Despite the large amount of research in FL for NLP, how different languages impact the FL training process has yet to be explored (Liu et al., 2021). Furthermore, multilingual FL provides an interesting and natural setting to explore non-IID data, of which different languages are an obvious example.

In this work, we explore multilingual federated learning across two multilingual language tasks and different stages of model pretraining. Our results show that fine-tuning pretrained models with FL can perform similarly to pretrained models fine-tuned with the standard centralized method (the no privacy setting), despite having completely non-IID language partitioned data. This finding shows

[1]Our code and data will be publicly released and available at <removed for anonymity>

* Authors contributed equally

that pretrained models provide an effective way for practitioners (and consumers) of multilingual data to gain the privacy benefits of FL at little or no cost to the final task performance.

2 Background and Related Work

The term *Federated Learning* was first proposed in McMahan et al. (2017), who applied the `FederatedAveraging` algorithm to the tasks of language modeling and image classification. Since then, much of the theoretical and applied work in FL (e.g. Chen et al. (2019b); Wu et al. (2020) among many others) has considered language modeling as a key task or benchmark.

Concurrent with the growing interest in Federated Learning, NLP has rapidly shifted toward the use of pretrained language models (PLMs) (e.g., BERT Devlin et al. (2019); T5 Raffel et al. (2019); GPT-3 Brown et al. (2020)). These PLMs are used for both the core task of next word prediction and as a starting point for learning other downstream NLP tasks. This *pretrain-and-fine-tune* paradigm has since become ubiquitous in modern NLP and has inspired a large and active area of research in model pretraining. Multilingual versions of these pretrained models have since been developed and are often used with transfer learning techniques to increase performance for tasks where data is limited (e.g. mBERT from Devlin et al. (2019)).

The intersection of distributed learning from private data partitions and PLMs is still a nascent area. Several works have explored more efficient methods of federated communication with the purpose of enabling these larger NLP models for production situations (Sui et al., 2020; Wu et al., 2021). Our work is orthogonal to these (and could be combined in future work), as we explore the effects of multilingual data on PLM FL, rather than creating methods to enable their use. Other papers focus on the gap between federated learning performance and centralized performance, evaluating on a wide variety of English NLP tasks (Liu and Miller, 2020; Lin et al., 2021; Chen et al., 2021). Although they focus on differential privacy (DP) rather than FL, Li et al. (2021) find that direct PLM training is difficult with standard DP methods, but that fine-tuning PLMs on English data is possible with private learning techniques. We differ from all these works by studying private learning, specifically FL, for PLMs in the novel multilingual setting.

3 Experimental Design

3.1 Federated Learning Methods

We use `FederatedAveraging` as the primary learning algorithm (McMahan et al., 2017). `FederatedAveraging` was introduced alongside the term Federated Learning and has been studied in both learning theory research (Stich, 2019) and applied work (Hard et al., 2018; Lin et al., 2021). In this algorithm, each client runs stochastic gradient descent (SGD) on its local data. After a specified number of steps, the client transmits its local model to the server, which averages these updates into a single centralized set of parameters. The server then broadcasts the centralized parameters to each client and the process repeats.

3.2 Client Partitioning

We consider three different training settings: standard training with no FL (e.g. *centralized* or *C*), FL with IID data (*FL IID* or *I*), where the data for each client is sampled randomly from all data, and FL with non-IID data (*FL non-IID* or *N*) where each client only sees data for one language (or for, one translation direction). See Figure 1 for a visual depiction of these three client partitioning schemes.

3.3 Data

We study two multilingual language tasks, due to their common use in the community: language modeling (LM) and machine translation (MT). We note that the data we use for training is relatively small; however, this mirrors real-life FL, as each client will not have a large amount of data in practical situations. We measure scores in perplexity (PPL) for LM and BLEU (Post, 2018) for MT.

Europarl We use the Europarl corpus (Koehn et al., 2005) taken from transcripts of European Union meetings. We sample data from eight languages: English, Spanish, Portuguese, French, German, Finnish, Polish, Lithuanian, and Czech. We sample 20k of each language for training and 5k for validation/testing, and use it for the LM task.

MTNT We use the Machine Translation of Noisy Text (MTNT) dataset (Michel and Neubig, 2018), which was the testset for the 2019 WMT robustness challenge. MTNT was gathered from user comments on Reddit discussion threads and contains noisy text including typos, casual language, and niche terminology. The dataset contains two non-English languages that we use: En $\rightarrow$ Fr and

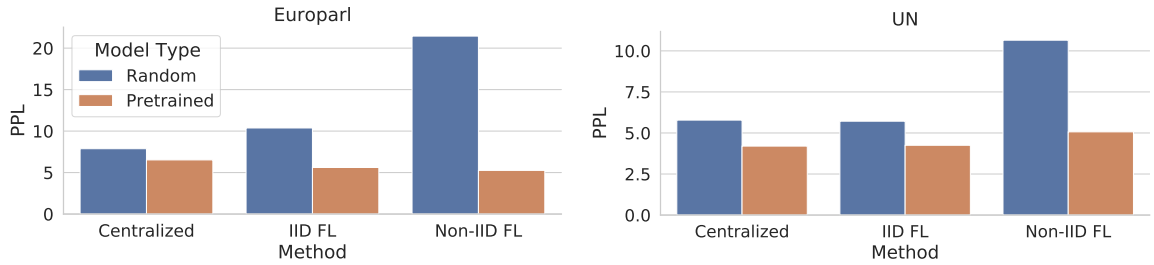

Figure 2: An overview of the language modeling results. Bars indicate the average language perplexity (PPL) over 8 languages for the Europarl dataset and 6 languages for the UN corpus. Lower is better.

	Europarl									UN						
M	**En**	**Cs**	**Lt**	**Es**	**Pl**	**Fi**	**Pt**	**De**	**Avg**	**En**	**Es**	**Fr**	**Ru**	**Zh**	**Ar**	**Avg**
C	**19.3**	**4.4**	**3.9**	**8.4**	**4.7**	**4.6**	**6.9**	**10.8**	**7.9**	**8.8**	**5.2**	**8.3**	3.8	4.2	**4.4**	**5.8**
I	27.1	5.3	4.4	11.2	5.8	5.4	8.7	15.1	10.4	**8.8**	**5.2**	8.6	**3.7**	**3.8**	**4.4**	**5.8**
N	50.4	6.8	12.3	16.1	18.3	11.3	34.6	21.8	21.4	12.3	11.4	14.8	9.1	8.0	8.3	10.7
C	12.0	**3.5**	**3.3**	13.4	4.7	**3.8**	**4.7**	6.8	6.5	6.7	4.1	**4.8**	2.9	**3.2**	3.5	**4.2**
I	10.5	3.9	4.2	6.1	**3.7**	4.4	5.4	6.7	5.6	**6.4**	**3.9**	5.8	**2.8**	**3.2**	**3.4**	4.3
N	**8.4**	3.7	4.0	**6.0**	**3.7**	4.3	5.5	**6.5**	**5.3**	6.9	4.5	6.4	4.2	4.3	4.1	5.1

Table 1: Results for FL experiments on the LM task. Bold scores indicate the best in the row for the given section. Scores are measured in perplexity (lower is better). The top rows are *from-scratch* models while the bottom rows use the pretrained version. Due to space we abbreviate: C for Centralized, I for IID FL, and N for non-IID FL.

En → Ja. This dataset has been used to test MT systems for robustness to domain shift (Li et al., 2019) and is suitable for our experiments since FL deals with client data that is uniquely shifted from centralized data. For more details on MTNT data preprocessing for M2M100, see Appendix C.

UN Corpus The UN Corpus (Ziemski et al., 2016) consists of official records from the UN proceedings over the years 1990 to 2014, in six languages: English, French, Spanish, Russian, Chinese, and Arabic. We use this data for LM (with 50k instances of training data per language and 5k for validation/testing) as well as three MT directions covering 6 languages (En → Fr, Ar → Es, Ru → Zh). Following previous work in MT adaption (see MTNT above) we sample 10k in each direction for training and 5k each for evaluation sets.

3.4 Modeling

For language modeling, we examine two different modeling settings: (1) fine-tuning pretrained multilingual models or (2) training the same multilingual model architecture but doing so with randomly initialized weights (also called training *from scratch*). For the MT experiments, we omit the *from-scratch* results as MT systems generally need large amounts of data to produce good results (see Appendix B for more details).

Our base model for the LM task is a distilled version of the mBERT model, shown to perform well on language modeling across many languages (Sanh et al., 2019; Devlin et al., 2019) while being smaller than the full mBERT.[2] For MT, we use the M2M-100 model (Fan et al., 2020) with 418 million parameters, a many-to-many MT model that can translate between any pairing of 100 languages. We note that although there are many more PLMs we could use, our overall results are consistent for both models and demonstrate that our conclusions are not specific to any one PLM.

3.5 Training

We use the Flower framework (Beutel et al., 2020) for federated training and evaluation due to its ease of use and strong community support. We use Hugging Face's *transformers* library (Wolf et al., 2019) for loading pretrained models and PyTorch as the underlying differentiation framework (Paszke et al., 2017). We train each LM model for 100 epochs if pretrained or 200 epochs if randomly initialized. For MT, we train for 25 epochs. For other hyperparameters and compute settings, see Appendix A.

[2]We note that mBERT uses masked language modeling (MLM) instead of standard language modeling, however, for the purposes of our analysis (as we do not seek to compare direct scores to previous work) MLM suffices. Furthermore, most multilingual PLMs train via some version of MLM.

Method	MTNT			UN			
	En-Fr	**En-Ja**	**Avg**	**En-Fr**	**Ar-Es**	**Ru-Zh**	**Avg**
No Training	30.8	14.2	22.5	31.4	27.5	15.2	24.7
Centralized	31.8	15.5	23.7	37.4	36.2	**22.6**	32.1
IID FL	**33.2**	15.6	**24.4**	**38.7**	**37.2**	21.4	**32.4**
non-IID FL	32.9	**15.7**	24.3	38.0	36.9	21.4	32.1

Table 2: Results for FL experiments on the Machine Translation task. Bold scores indicate the best in the row. Scores are measured in BLEU (Post, 2018), higher is better.

4 Results

Language Modeling In Figure 2 we see the overall results of the language modeling task across the two datasets. As expected, the from-scratch models perform much worse than the pretrained models. Interestingly, however, the gap between FL and centralized methods decreases when using pretrained models (especially so for the non-IID setting), indicating that pretrained models are an effective method for federated learning.

In Table 1 we show results broken down by language. Again we see that in the from-scratch category the centralized model is the same or better than the FL methods in almost every single language, across both datasets. However, in the pretrained section the results are more mixed, with the centralized model winning or tying in 4 of the 8 Europarl languages and 2 of the 6 UN languages.

Examining the difference between IID FL and non-IID FL, we see that the IID FL method is better in three of the four settings (e.g. Europarl Random: 21.4 non-IID vs 10.4 IID). However, we also see that the gap between IID and non-IID FL scores is smaller in the pretrained setting (Table 2, bottom) showing again the effectiveness of pretraining.

Machine Translation Table 2's results show similar conclusions to those of the Language Modeling task. We see that on the MTNT dataset, the FL algorithms actually outperform centralized learning (24.4 avg. BLEU for IID FL vs 23.7 for Centralized). While the non-IID FL setting is below IID performance on En-Fr, it has the highest score on the difficult En-Ja setting (15.7 BLEU).

On the UN corpus, we see that again the IID FL model performs best, although all three models perform similarly (with non-IID tying Centralized).

To examine how well these models learned compared to the original multilingual model, we can look at the first row of Table 2. In every case, all learning algorithms improve over the baseline.

Discussion Our examination of multilingual FL indicates that the gap between federated and centralized learning is much smaller or near zero when pretrained models are used. Despite the fact that local models are averaged together, even non-IID data partitioning (where each client sees only one language) has a small impact on final multilingual performance, when using pretrained models. These findings suggest that, when possible, practitioners who need multilingual federated learning should employ pretrained models in order to gain the privacy benefits of federated learning, without taking much (if any) of a performance loss to do so.

Furthermore, our analysis shows that these findings hold for different multilingual models, on disparate NLP tasks, and across 13 different languages. We acknowledge that the languages used in this study are generally considered higher-resource, but expect that these conclusions will continue to hold as long as the pretrained model is effective on the target language (or language pairs, for MT).

5 Conclusion

In this work we provided the first analysis of multilingual language data on federated learning algorithms. We found that fine-tuning a pretrained model with FL methods can yield similar performance to centralized learning, even when clients are partitioned by language (non-IID FL). However, models trained from scratch still show a large gap between centralized and federated learning. Our results suggest that learning on private partitioned data is possible without having to incur a large performance penalty. We hope that these results will aid practitioners in using FL (and also downstream consumers) and inspire the broader community to consider multilingual data in future federated learning research for natural language processing.

References

Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. 2020. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2021. Fedmatch: Federated learning over heterogeneous question answering data. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

Mingqing Chen, Rajiv Mathews, Tom Ouyang, and Françoise Beaufays. 2019a. Federated learning of out-of-vocabulary words. *arXiv preprint arXiv:1903.10635*.

Mingqing Chen, Ananda Theertha Suresh, Rajiv Mathews, Adeline Wong, Cyril Allauzen, Françoise Beaufays, and Michael Riley. 2019b. Federated learning of n-gram language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 121–130.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Çelebi, 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. *ArXiv*, abs/2010.11125.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, and Zi Huang. 2019. Learning private neural language modeling with attentive aggregation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn et al. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Jakub Konečnỳ, 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*.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Pino, and Hassan Sajjad. 2019. Findings of the first shared task on machine translation robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 91–102, Florence, Italy. Association for Computational Linguistics.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2021. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*.

Bill Yuchen Lin, Chaoyang He, ZiHang Zeng, Hulin Wang, Yufen Huang, M. Soltanolkotabi, Xiang Ren, and S. Avestimehr. 2021. Fednlp: A research platform for federated learning in natural language processing. In *arXiv cs.CL 2104.08815*.

Dianbo Liu and Tim Miller. 2020. Federated pretraining and fine tuning of bert using clinical notes from multiple silos. *ArXiv*, abs/2002.08562.

Ming Liu, Stella Ho, Mengqi Wang, Longxiang Gao, Yuan Jin, and He Zhang. 2021. Federated learning meets natural language processing: A survey. *arXiv preprint arXiv:2107.12603*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. 2019. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sebastian Urban Stich. 2019. Local sgd converges fast and communicates little. In *ICLR 2019-International Conference on Learning Representations*, CONF.

Asa Cooper Stickland, Xian Li, and Marjan Ghazvininejad. 2020. Recipes for adapting pre-trained monolingual and multilingual models to machine translation. *arXiv preprint arXiv:2004.14911*.

Joel Stremmel and Arjun Singh. 2020. Pretraining federated text models for next word prediction. *arXiv preprint arXiv:2005.04828*.

Dianbo Sui, Yubo Chen, Jun Zhao, Yantao Jia, Yuantao Xie, and Weijian Sun. 2020. Feded: Federated learning via ensemble distillation for medical relation extraction. In *EMNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Chuhan Wu, Fangzhao Wu, Ruixuan Liu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2021. Fedkd: Communication efficient federated learning via knowledge distillation. *ArXiv*, abs/2108.13323.

Xing Wu, Zhaowang Liang, and Jianjia Wang. 2020. Fedmed: A federated learning framework for language modeling. *Sensors*, 20(14):4048.

Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534.

A Hyperparameters

Each LM experiment ran for approximately a day each on a 5 GPU cluster of RTX 6000 GPUs with 24GB of memory per GPU. The MT experiments took approximately 12 hours each on the same cluster.

We use the AdamW optimizer (Loshchilov and Hutter, 2017; Kingma and Ba, 2014) for all experiments (shown to be effective for FL in Lin et al. (2021)). Each client goes through a full epoch of local learning before synchronizing with the server.

For MT, we report results using the 5e-5 learning rate, as we found in initial results (as have others also, see Appendix B of Stickland et al. (2020) as one example) that MT experiments are generally consistent over learning rates when fine-tuning. For language modeling, we use three different learning rates (1e-4, 5e-5, 1e-6). All models were selected using the best performing version on the validation set, for the given model and training setting. For both tasks, we use early stopping (5 epochs of no improvement for MT, 10 for LM).

We use the standard Sacrebleu settings: nrefs:1, mixed case, eff:no, tok:13a, smooth:exp, and version 2.0.0. For Ja and Zh we use their respective tokenizers.

B From Scratch MT

We do not report results for from-scratch training of MT systems, as large neural MT systems generally need large amounts of data to be effective. We ran experiments for the MTNT dataset from scratch, running for twice as many epochs. Resulting models appeared to converge by loss but had extremely low BLEU scores. Thus, we only include pretrained results in Table 2.

C MTNT Data Preprocessing for M2M100

M2M100 was trained using scripts that removed input with "excess punctuation." We follow this in preparing MTNT training data. We use all En $\rightarrow$ Ja data (consisting of approximately 6k instances) and take the corresponding En $\rightarrow$ Fr instances, randomly sampling additional instances until there are the same number of instances in each direction. We sample an equal number of training instances as we are testing the effects of multilingual data, rather than unequal dataset sizes. We then remove the training instances with excess punctuation (or sentences less than 3 characters) following the M2M100 script. This leaves 5605 instances in each direction for training. We use the standard MTNT dev and test sets, as-is, consisting of approximately 1k data points.