Too Brittle To Touch: Comparing the Stability of Quantization and Distillation Towards Developing Lightweight Low-Resource MT Models

Harshita Diddee[1] **Sandipan Dandapat**[2] **Monojit Choudhury**[1]
Tanuja Ganu[1] **Kalika Bali**[1]
[1] Microsoft Research, India
[2] Microsoft R&D, India
{t-hdiddee,sadandap,monojitc,taganu,kalikab}@microsoft.com

Abstract

Leveraging shared learning through Massively Multilingual Models, state-of-the-art machine translation (MT) models are often able to adapt to the paucity of data for low-resource languages. However, this performance comes at the cost of significantly bloated models which are not practically deployable. Knowledge Distillation is one popular technique to develop competitive lightweight models: In this work, we first evaluate it's use to compress MT models focusing specifically on languages with extremely limited training data. Through our analysis across 8 languages, we find that the variance in the performance of the distilled models due to their dependence on priors including the amount of synthetic data used for distillation, the student architecture, training hyper-parameters and confidence of the teacher models, makes distillation a brittle compression mechanism. To mitigate this, we explore the use of post-training quantization for the compression of these models. Here, we find that while distillation provides gains across some low-resource languages, quantization provides more consistent performance trends for the entire range of languages, especially the lowest-resource languages in our target set.

1 Introduction

While NLP has made giant strides in producing more accurate models, these benefits are often not transferred representatively to end-users who would eventually use a language-technology (Ethayarajh and Jurafsky, 2020; Caselli et al., 2021). Bloated sizes, cumbersome inference times (Tao et al., 2022a) and a limited set of languages that these models serve are a few reasons for this. More specifically, their usage is hindered by access bottlenecks such as (a) **Infrastructural Obstacles**: A large percentage of end-users do not have sustained access to internet or high-compute devices to enjoy a stable access to cloud-inferencing of current NLP models (Ranathunga and de Silva, 2022; Diddee et al., 2022), (b) **Latency Requirements**: Certain NLP services (chat-bots, real-time assistance interfaces, etc.) require very low-inference time which requisite lightweight-models (c) **Privacy Constraints**: The outflow of sensitive user data which is fed for inferencing to remotely hosted NLP models also has well documented issues (Srinath et al., 2021; Huang and Chen, 2021; Huang et al., 2020; Diddee and Kansra, 2020).

Within the research that focuses on evaluating and mitigating these practical constraints, the focus on low-resource language setups has been fairly limited (Ganesh et al., 2021). For instance, while the compression of large language models has received consistent attention through analysis of pruning (Behnke and Heafield, 2020; Behnke et al., 2021), distillation (Bapna et al., 2022; Mghabbar and Ratnamogan, 2020; Kim and Rush, 2016; Junczys-Dowmunt et al., 2018) and even quantization (Bondarenko et al., 2021; Zadeh et al., 2020) - much of this work has focused on compressing language models for high-resource languages.

In this paper, we report the results of a comparative analysis of the performance of distillation and quantization. By focusing on compressing seq2seq multilingual models across a range of languages with data ranging from 7000 to 3M samples - we especially demonstrate the different priors that need to be ascertained for the successful distillation of the model. We are unaware of any previous study that demonstrates the performance of these mechanisms on such low resource languages.

The utility of this work is in commenting on the feasibility of these two compression techniques for rapid development and deployment of MT Models for low resource languages (Joshi et al., 2020). More specifically, we believe that distillation's reliance on several priors can be addressed naively through a resource-intensive exercise, where the optimal values of these priors are computed exhaustively. However, in the absence of such a budget,

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 870–885
Abu Dhabi, December 7–8, 2022.

we expect this to be a major impediment in the development of lightweight models for such languages. Since low resource language communities may also be marginalised in other ways, exhaustive investment of data and compute might not be feasible for such communities as well as the language technologists working on these languages (Zhang et al., 2022; Diddee et al., 2022; Markl, 2022).

The main contributions of this work are:

1. We distill competitive baseline models for 8 low-resource languages (Bribri, Wixarica, Gondi, Mundari, Assamese, Odia, Punjabi and Gujarati) and evaluate the sensitivity of the generated models to priors including (a) amount of synthetic Data being used for training (b) The architecture of the student model (c) the training hyper-parameter configuration and (d) the confidence of the teacher models.

2. We, then, quantize these models to observe if quantization provides a more consistent compression mechanism for these languages. Based on our analysis, we conclude that the suprising stability of naive Post-Training Quantization, especially in the compression of extremely-low resource languages (training data between 5000 and 25000 samples) over distillation.

We release a combination of lightweight, offline support MT models for these languages along with the scripts for generation and offline inference to further reproducible research in this domain[1].

2 Approach - Model and Size Adaptations

In this section, we describe the languages (2.1), architectures under consideration (2.1), the adaptations that we make for training and fine-tuning these models (2.2) and the adaptations we make to compress their size.

2.1 Languages

We perform our analysis on the eight languages shown in Table 1. These languages cover a wide range of availability of monolingual and parallel data, spanning from classes 0 to 3 as defined in Joshi et al. (2020). Additionally, they differ in scripts and their inclusion in pretraining corpus which result in interesting modelling adaptions that are needed to be performed for the development of their baselines. In this work, we only study the *High-Resource Language* (HRL) $\rightarrow$ *Low-Resource Language* (LRL) translation direction. The source languages for all our target languages are mentioned in Table 1.

Family of Models For this work, we leverage two model classes to carry out our analysis: **I)** seq2seq transformer (Vaswani et al., 2017), hereafter referred to as vanilla transformer: With 6 Encoder and Decoder Layers, Vocabulary size - varying between 8k to 32k and 8 attention heads. and **II)** mT5-small (Xue et al., 2021): With 8 Encoder and Decoder Layers, Vocabulary Size - 250100 and 6 attention heads.

We train the vanilla transformer from scratch, hereafter referred to as transformer, to develop a naive baseline for our experiments, and further fine-tune the mT5-small, hereafter referred to as mT5, with certain adaptations for all the languages, as discussed in section 2.2.

For ease of reporting, we define the highest-performing-model (denoted by HM) over our family of models as:

$$\mathrm{HM} = \underset{\mathrm{M}}{\operatorname{argmax}}\, \mathrm{A}(\mathrm{M})$$

where M is a model class with performance $\mathrm{A}(\mathrm{M})$ after training (where A is a metric like BLEU (Papineni et al., 2002) or chrF (Popović, 2016) used to monitor the task-specific performance of the model).

2.2 Model Adaptations: Language Specific Approaches

Here we describe the strategies required to adapt these models to different low-resource languages: During fine-tuning, we adapt the pretrained mT5 tokenizer to unseen scripts (encountered for Odia) by transliterating it to the closest, highest-resource language included in the pretraining corpus of the pretrained model (Khemchandani et al., 2021; Ramesh et al., 2021, 2022). For our extremely low-resource languages, we used Lexicon-Adaption (Wang et al., 2022) for the augmentation of target-side monolingual data for languages wherever a bilingual lexicon could be leveraged - Detailed performance with Hindi-Gondi is provided in the Appendix section A.2. However since such methods were not extensible to all the languages in our target language set, we report final experimental results on the models

[1] Codebase and Open-Sourced Models

Language	Class	Source Language	Data Constraints		Model Constraints	
			Monolingual Data	Parallel Data	Shared Script	Included in Pretraining
Bribri	0	Spanish	✗	✓	✗	✗
Wixarica	0	Spanish	✗	✓	✗	✗
Mundari	0	Hindi	✗	✓	✓	✗
Gondi	0	Hindi	✗	✓	✓	✗
Assammese	1	English	✓	✓	✓	✓
Odia	1	English	✓	✓	✗	✗
Punjabi	2	English	✓	✓	✗	✓
Gujarati	1	English	✓	✓	✗	✓

Table 1: Languages Under Consideration: Note that the except the language's inclusion in the pretraining corpus of our chosen pretrained language models, all factors are independent of our experimental setup. Source language column enlists the source language of the translation pairs

which did not leverage any additional data other than the data mentioned in A.1. Since we analyze the HRL to LRL direction and 4 out of 8 (Bribri, Wixarica, Gondi and Mundari) of our target languages have little to negligible monolingual data - we were also unable to leverage Back-Translation to augment our language-specific parallel corpus (Edunov et al., 2018).

2.3 Size Adaptation: Knowledge Distillation

Knowledge distillation involves training a smaller student network to mimic the token level probabilities of a larger, more accurate teacher model. We distill our models using Hard Distillation (Kim and Rush, 2016): we utilize a set of monolingual sentences in the HRL - and forward translate using the HM to generate synthetic labels that a lighter student model is then trained on.

2.3.1 Estimation of Optimal Values for Priors

We define a prior as any attribute of the compression mechanism that needs to be initialized meaningfully and/or optimized for optimal performance - akin to hyperparameters. We use this term specifically so as to put all the dependent variables - such as training data, prediction confidence of the uncompressed models, etc in a single bucket: rather than using a term like hyperparameters that already holds traditional significance in literature. The experimental sweeps for these priors are briefly explained in this section. Note that we focus largely on distillation while estimating for these priors, because quantization provides competitive models even with the default choices established by literature whereas with distillation - the estimation of these priors is critical to achieve a competitive compressed model variant in most cases.

Prior 1: Optimal Student Architecture Following prior work like Bapna et al. (2022), we experimented with 3 candidate architectures, two of which used deep encoders and shallower decoders. We sweeped across 3 candidate architectures - all variants of a seq2seq transformers with (a) 8 Encoder + 6 Decoder Layers (b) 6 Encoder + 4 Decoder Layers and (c) 6 Encoder + 3 Decoder Layers. We chose the architecture that gave the best BLEU performance after 30 epochs. Sweeps for the architecture were done across each of the following languages - Gondi, Assamesse and Odia as they covered a wide range of training data.

Prior 2: Optimal Training Hyperparameters We sweeped across a set of hyper-parameter sets for Bribri, Gondi, Assamesse and Gujarati to identify the optimal set for the distilled student models. Our goal here was to specifically study the transferability of a hyperparameter set which performed competitively for one or more languages, to all the languages in our target set.

Prior 3: Amount of Training Data for the Student We sweeped across 3 candidate sizes of our synthetic dataset: 100K, 250K and 500K pseudo-labels. Since this decision could also be greatly dependent on the quality of the labels generated per language - we ran this sweep for Bribri, Gondi, Odia and Gujarati, as the quality of the labels generated by the teachers for these languages would be expected to demonstrate significant variation.

Prior 4: Optimal Teacher Architecture To do a preliminary quantification of the effect of the choice of a teacher architecture and the quantity of data that a teacher is trained for on the compressibility of the model - we decided to evaluate the confidence of our teacher models on the predictions they generated. For this, we sampled 100 instances

from each of our testsets and monitored the logit distribution of our teacher models. Specifically, we calculated the average of the softmax entropy of the token-level softmax distributions for a sequence. Taking inspiration from the unsupervised estimation of quality of machine translation outputs (Fomicheva et al., 2020) through similar methods, we hypothesised that the lower the entropy of our model, the more confident it would be in its predictions for a given sample. The intuition here was that if a model is confident about its prediction, its logit distribution would be highly-skewed, and not resemble a uniform distribution (which would indicate its indecisiveness in being able to predict the right token - and therefore, the right sequence). Eventually, this could be used to gauge the quality of the pseudo labels that are student were being trained on.

2.4 Size Adaptation: Quantization

Quantization is a common way to reduce the computational time and memory consumption of neural networks (Wu et al., 2020). Here, a lower-bit representation of weights and activation functions is used to achieve a lower memory footprint. In this work, we perform post-training quantization, where after training the base model with full precision of floating point 32 bits (fp-32), we convert the weights and activations of the model to 8 bit integers (int-8). Note that during inference, we still preserve the precision of the input and output encoder-decoder distributions as fp-32. In theory, this brings down the memory consumption of the model by nearly 4x times, though we see an effective reduction of about 3x in practice. More details on the memory-reductions achieved are specified in the Appendix A.4

3 Experimental Setup

3.1 Data

(a) Bribri and Wixarica: We use the training data 7K and 8K sentences, respectively from Feldman and Coto-Solano (2020) and evaluate on test data from Mager et al. (2021). **(b) Gondi**: We use 26k sentences from the data opensourced by CGNET Swara (CGNET, 2019) and split it into training and test sets.[2] **(c) Mundari:** We use a dataset of 10K sentences provided by Indian Institute of Technology, Kharagpur[3], and split it into training and test sets.[1] **(d) Assamesse, Odia, Punjabi and Gujarati**: We use the training data from Ramesh et al. (2022) (with 0.14M, 1M, 2.4M and 3M sentences, respectively) and evaluate on test data from FLORES200 Goyal et al. (2022) for Assamese and WAT2021 Nakazawa et al. (2021) for the remaining languages. Additional details about datasets (sizes and splits) are mentioned in the Appendix A.1.

3.2 Training Setup

Hyperparameters: We use the transformer and mT5 as our model classes as described previously in Section 2. The hyperparameters for our transformer model was optimized for fine-tuning of Odia, trained on 1M sentence pairs. For fine-tuning, we use the Adafactor optimizer (Shazeer and Stern, 2018), with a linearly decaying learning rate of 1e-3. Since training with smaller batches is known to be more effective for extremely low-resource language training (Atrio and Popescu-Belis, 2022), we tuned the training batch size for every language - varying from 32 to 256 (with gradient accumulation as 2) though we did not see very significant variation in the performance on the basis of this tuning. For our stopping criteria: we fine-tuned all models for 60 epochs (which concluded with considerably overfit models) and then selected models by we picking the checkpoint which had the best validation performance on BLEU (with only the 13a tokenizer which mimics the mteval-v13a script from Moses) (Post, 2018).

We use the sentencepiece tokenizer to build tokenizers for training the baselines for each of the languages (Kudo and Richardson, 2018). We use the per-token cross-entropy loss for fine-tuning all our models. Following Xu et al. (2021), we opt for a relatively smaller vocabulary size with the intent of learning more meaningful subword representations for our extremely low-resource languages. Specifically, we use a vocabulary size of 8K for Gondi, Mundari, Bribri and Wixarica, compared to 32K used for Assamesse, Odia Punjabi and Gujarati.

Experimental Setup for Distillation For Mundari and Gondi we utilize 500K Hindi sentences sampled from the Samanantar corpus (Ramesh et al., 2022); We use the corresponding English corpus to sample English sentences for generating the pseudo labels for Assamesse, Odia,

[2]To avoid any test-set leaks, we deduplicate the data by removing tuples (S^i, T^i) where S^i is the i^{th} sentence in the source language and T^i is $i^{th}$the sentence in the target language, between the train and the test set.

[3]Data to be released soon;

Punjabi and Gujarati. For Bribri and Wixarica - We use Spanish data made available by the Tatoeba Challenge (Tiedemann, 2020). We use the per-token cross-entropy loss for training our distilled models.

Evaluation Metrics: We use BLEU (sacrebleu with spm pre-tokenization (version 2.2.0)) (Post, 2018) for all our evaluations (Goyal et al., 2020). In addition to this, we also report chrF2 (Popović, 2016) for all our experiments for a more comprehensive comparison between the models.

4 Results

In section 4.1, we present the performances of our base models in Table 2. In the following section 4.2, we report the performances of the distilled HM in Table 3. Using these empirical results we focus on answering the following questions (a) To what degree can scaling the student training data improve the performance of the student model? (4.3) (b) How sensitive is distillation to the choice of the architecture of the student model? (4.4) (c) How can we choose an optimal teacher that is most suitable for compression? (4.5) (d) To what degree does the hyperparameter set suitable for distilling a model for one language transfer to another language? (4.6)

While answering these questions, we also analyze in parallel the performance of the quantized variants of these models implicitly indicating the reduced sensitivity of these variants from most of the previously discussed priors in spite of their competitive performances.

Language	Data	Vanilla transformer		mT5	
		spBLEU	chrF2	spBLEU	chrF2
Bribri	7K	1.7	11.6	**6.4**	19.3
Wixarica	8K	2.2	14.1	**6.2**	28.0
Mundari	10k	0.1	5.6	**15.9**	33.7
Gondi	26K	1.2	7.9	**14.3**	32.5
Assamesse	140K	0.8	12.4	**10.7**	30.4
Odia	1M	23.7	43.6	**27.4**	47.6
Punjabi	2.4M	**38.4**	50.6	34.8	44.1
Gujarati	3.05M	**35.9**	53.4	35.7	49.8

Table 2: Performance of our base models (transformer and mT5) without quantization or distillation. Best performing models out of the two architectures are marked in bold.

4.1 Analyzing the Baseline Models

As expected, the transformer models for target languages start competing (and outperforming) once an adequate amount of data is available for training the vanilla transformers. In addition to the obvious gain for being only optimized for target languages, the performance gains of these baselines can also be attributed to the language-specific tokenizer that they utilize, in contrast to the pretrained mT5 tokenizer that might be sub-optimal for language-specific generation. For our low-resource languages though, the advantage of transfer learning is clearly evident: all languages achieve a minimum and maximum performance improvement of 4 and 16 BLEU points. Gondi and Mundari, despite having relatively low-amount of data, perform well - though we expect an overestimation of their performance due to the homogenity between the train and the test set. Additionally though, both languages share scripts with a dominant language script i.e., Devanagari and hence, can be expected to gain because of that.

4.2 Analyzing the Compressed Models

In Table 3, we briefly present the performances of our distilled and quantized models. As evident, especially for the lowest-resource models, both distillation and quantization give competitive performance in addition to providing a significant size reduction. Note that Table 3 does not report the performance of the quantization of the vanilla transformer models for Odia, Gujarati and Punjabi even though they had competed or outperformed the mT5 variants. This is because they suffered a significant drop in performance - Odia dropped in performance to 8.4 BLEU/30.5 chrF2 in contrast to its HM scores of 23.7 BLEU/ 43.6 chrF2 respectively. Gujarati and Punjabi also dropped to 16 BLEU/31.2 chrF2 and 19.1/36.0 , respectively. To explain this we note what distinguishes these two architectures: (a) mT5 is deeper than transformer having 2 extra layers on the encoder's side than the vanilla transformer and (b) leverages multilingual pretraining. These attributes become useful in interpreting mT5 robustness to compression. In agreement with prior work like Li et al. (2020), deeper models can be expected to be more immune to compression. In fact, these models can be expected to be regularized by a certain degree through quantization, and we posit that we might be adopting a sub-optimal fine-tuning hyperparameter set for the initial fine-tuning of these models, consequently generating potentially overfit models and this gets mitigated to some extent upon quantiza-

tion. Taking into consideration the lack of prior work on fine-tuning large LMs on such extremely low-resource languages and the infeasibility of running intricate hyperparameter sweeps per language with such large models, this can also be expected to degrade the quality of the labels generated for training the distilled models - ultimately affecting the performance that the distilled models achieve.

Language	HM	Distilled HM		Quantized HM	
	spBLEU	spBLEU	chrF2	spBLEU	chrF2
Bribri	6.4	6.8	13.2	**7.4**	19.4
Wixarica	6.2	4.1	17.3	**7.2**	26.8
Mundari	15.9	**18.2**	32.7	15.7	29.3
Gondi	14.3	**14.2**	32.8	13.8	31.1
Assamesse	10.7	**9.6**	27.4	6.2	25.7
Odia	27.4	20.2	40.7	**21.0**	41.3
Punjabi	38.4	**32.8**	46.6	27.0	48.0
Gujarati	35.9	**29.8**	48.6	28.4	51.4

Table 3: Performance of the HM for all languages after applying Distillation and Quantization. Best performing models out of both of the size adaptations are marked in bold.

In the following sections we focus on presenting our analysis of distillation's sensitivity to certain priors. In each section, we also discuss an analysis of the same priors' effect on quantization. Note that since the mT5 outperformed the vanilla transformer variants for all languages up till Odia - we distilled and quantized them for these languages. Also note that the HM for these languages is hence, mT5. Additionally, for Odia, Gujarati and Punjabi, we quantized both the mT5 and the vanilla transformer variants of the models.

4.3 Sensitivity to Priors: Data

The quality, quantity and the domain of data that the teacher or uncompressed variant of the model is trained on, appears to impact both the mechanisms of compression: For distillation the gold training data as well as the monolingual data utilized for generating student labels is of relevance, and for quantization only the gold data that the teacher is fine-tuned for, is of relevance.

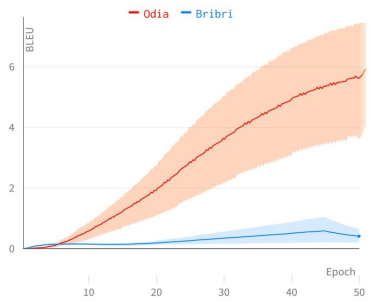
Quantity of Training Data Interestingly, quantization displayed consistent performance variations across the entire range of our low-resource language sets (all languages up till Odia), giving marginally close scores to the HM so at least within the data sparse languages we did not see any direct variation in the performance according to the amount of training data used. Both mechanisms show nearly equal degradation in performance for the HRL.

Quality of Training Data The quality of the data that the teacher is trained on affects the model's immunity to compression. This is best demonstrated by the post-compression performances of Gondi and Mundari in Table 3: In Gondi - the train set has nearly 26K sentences, which by the virtue of being collected via crowd-sourcing may be expected to be noisy. Mundari's training data, though also crowd-sourced, claims to have been validated manually after its collection by the providers to generate the final corpus of about 10K sentences. The observed difference where Gondi suffers a slight performance degradation post-compression and Mundari experiences a significant performance gain, may be attributed to the difference in the quality of their training data. Note that both languages are being translated from the same source language, share the same script and are being tested on a correlated test set - so the quality and quantity of training data are expected to be major contributors to the variations in their performance.[4]

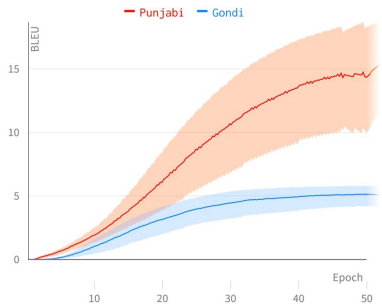
Quantity of Pseudo-Labels used for Student's Training Results of our analysis of scaling student data between 100K to 500K are presented in Figure 1. More data seemed to help for the entire spectrum of languages - though it is evident that the gain in the performance diminished in proportion to the amount of added data as we approached the lowest-resource languages in our set. The gain in performance upon the addition to 250K samples to a HRL like Odia or Punjabi is significantly more pronounced than the gain in performance for Bribri or Gondi - where there is a very marginal improvement in the performance upon the addition of 250K samples. This could be indicative of the diminishing efficacy of the increasingly noisy data that was generated by the lowest-resource teachers. We explore this notion in more depth in Section 4.5.

Domain of Data While we do not perform any targeted experiments to evaluate the domain dependence of the two compression mechanisms - we posit that the distilled models' significantly better performance than its quantized variant in As-

[4]The two languages do belong to two different language families - Gondi belonging to the Dravidian language family which has a higher representation in the pretraining corpus for mT5, and Mundari being Austro-Asiatic



(a) Variation in the efficacy of pseudo-labels between Bribri and Odia



(b) Variation in the efficacy of pseudo-labels between Punjabi and Gondi

Figure 1: Min/Max range curves of the performance of the models trained on scaled data: The shaded range is considerably lower for the lowest-resource languages indicating reduced efficacy of scaling student data.

samesse could be attributed to the distilled model's exposure to the diverse-domain data during the student's training. Note that the testset used in Assamesse, FLORES 200 (Goyal et al., 2022), is claimed to be of a very diverse-domain origin. Given this, the process of training a student on monolingual data of a potentially more diverse origin to that of the native training set - would explain the gain that the language demonstrates on a domain-agnostic testset. Prior work like Mghabbar and Ratnamogan (2020) already shows distillation's efficacy in enabling students to adapt to out-of-domain data that the teacher may not have ever been exposed to. Quantization on the other hand, has no opportunity for exposure to any out-of-domain data - so its adaptation and performance across a domain-agnostic testset can be expected to only degrade.

4.4 Sensitivity to Priors: Student Architecture

We find that distilled student models could be adversely sub-optimal for a given language, despite being sub-optimal or even an optimal choice for a large subset of languages. To demonstrate this

Figure 2: Variation in BLEU due to difference in the choice of a student architecture: An optimal architecture choice for Odia and Gondi gives adversely sub-optimal performance for Assamesse

in Figure 2, we show the performance of two distilled models on an identical hyperparameter set and student architecture. While the chosen student architecture gives competitive performances for Gondi and Odia, Assamesse performs significantly worse for this candidate architecture. We did attempt retraining the model with a different seed to negate the possibility of a randomly poor initialization though this did not improve the convergence. While we did not notice such a drastic performance variation across any other candidate set, this instance did indicate brittleness to the student-architecture for a given language. After these sweeps, we fixed a transformer-based encoder with 6 layers and a transformer-based decoder with 4 layers as the distilled model for our further experiments.

4.5 Sensitivity to Priors: Confidence of the Teacher Model

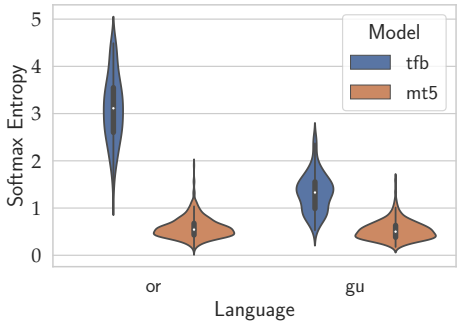


Figure 3: Entropy distributions of mT5 and transformer: lower-entropy indicates high-confidence and consequently suggest higher-quality of translations.

Estimating the confidence of our teacher models displayed manifold benefits: Within Distillation, it helped us get an indirect estimate of the qual-

ity of the training data that the student model was trained on. Within Quantization, it was useful in analyzing why the mT5-variants were more robust to quantization. Note that since the testsets for all the languages are of varying difficulty - doing a language-wise comparison on the basis of such metrics was non-trivial since the confidence predictions could also vary in accordance with the complexity of the testsets being evaluated upon. Hence, we majorly focused on analyzing languages which were either evaluated on the same test set (Gujarati, Punjabi, Odia with WAT21 testset (Nakazawa et al., 2021)) or the different architectures for each of our languages which could be evaluated for the same testset.

Figure 3 demonstrates the difference in the entropy of the softmax distributions of the mT5 and transformer teacher variants. Note that this is for Gujarati and Odia, our highest resource language, for which both architectures perform quite competitively and the vanilla transformer even outperforms the mT5.

As is evident, the mT5 variant has much lower entropy scores, with lower dispersion indicating high-confidence in the predictions it produces for each of the samples. Note that the inference pipeline for both architectures is identical - Greedy Search with no sampling so we don't expect any difference in the decoding mechanism to affect the quality or distribution of representations that we are monitoring. This is a very interesting observation, as both models appear to perform comparably according to our automatic metric evaluations - yet differ quite significantly in the stability with which they generate these predictions.

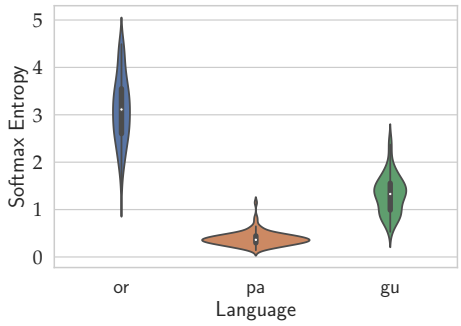


Figure 4: Entropy distributions for transformer across different languages: Models become increasingly more confident about their predictions with an increase in training data

Next, we attempt to establish if training with more data makes a model more confident in its prediction. Figure 4 demonstrates the entropy scores for Odia, Punjabi and Gujarati. Each of these have data increasing in the order of 1M, 2.4M and 3M respectively. Here we observe that indeed, models trained with more data achieved consistently lower entropy scores.

4.6 Sensitivity to Training Hyperparameters

In this section we present results of evaluating if an adequate hyperparameter set for a given language may be suitable for generating an optimal variant for another distilled language. Here too, we demonstrate using a subset of our hyperparameter sweep that there can be a marked degradation in the suitability of an averagely optimal hyperparameter set (that might be close to optimal to multiple languages with similar attributes) to an unseen language;

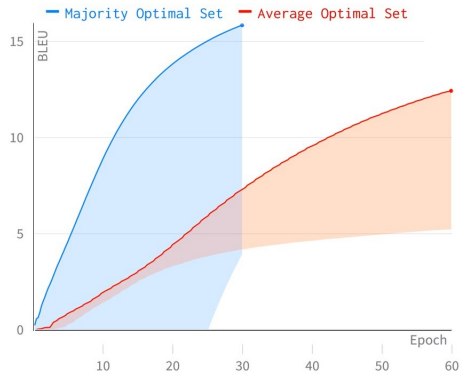


Figure 5: Min/Max range of performances of Gujarati, Bribri and Assamesse across a hyperparameter set that is optimal for these languages but adversely sub-optimal set for Gondi

In Figure 5, when tuned for the hyperparameter set that is optimal for a majority of languages in our set, Gondi does not even converge as a result of which the lower-bound of a teacher's performance for that hyperparameter set is 0. Note that this hyperparameter set transferability does not seem to show any specific data oriented trends as well. For instance, the same hyperparameter set that was optimal for Gujarati, our highest resource language with 3M data points, is only slightly sub-optimal for Bribri, our lowest resource language with 7000 data points, and Assamese, our mid-resourced language with 135K sentences. Also note that we were able to get acceptable performance for Gondi with almost an identical hyperparameter setup with a larger batch size (quadrupled to the one in this

setup) indicating that a per-language sweep would be an ideal and acceptable solution even though this would imply that distilling models would mandate a significant hyperparameter tuning for achieving optimal performance. A detailed list of what hyperparameters we sweeped through can be found in the Appendix Table 5.

5 Takeaways

We encapsulate the learning from our analysis as the following takeaways:

1. ***Data Dependence of the Method of Compression:*** Training teacher models with lesser quantity, higher quality data is expected to improve a model's robustness against both quantization and distillation. The post-quantization performance suffers equally for models trained with varying degrees of data. This is not the case with distillation, where increasing the amount of training data for student distilled models starts providing diminishing returns as the amount of training data for the teacher reduces.

2. ***Cost of Compression:*** Distillation is quite sensitive to its training hyperparameters and the student's architecture. This choice doesn't necessarily follow any data-oriented trends as well i.e., languages having similar amount of data may perform very differently on similar hyperparameter and student architecture sets. Hence, Distillation mandates a significant hyperparameter tuning cost that Quantization does not incur.

3. ***Stability of Compression*:** Hard Distillation and Post-Training Quantization are both promising methods of quickly compressing massively multilingual models for machine translation for extremely low-resource languages. Post-Training Quantization should be preferred when the uncompressed variants is pretrained and/or deep, expected degree of compression is upto 4x the original model's size and the cost of compression is to be minimum. Distillation, on the other hand, should be preferred when domain-expansion, language-specific tokenization and more than 4x degree of compression needs to be achieved at the cost of a tuning for optimal architecture and training setup selection.

6 Related Work

Owing to the known benefits of compressing language models due to their lower-memory footprint, improved inference speed and even improved performance in some cases, compression techniques have been explored widely in NLP.

Quantization While the work on quantizing encoder-models is replete (Zafrir et al., 2019; Bondarenko et al., 2021; Kim et al., 2021; Zadeh et al., 2020) the focus on quantizing decoder-only models (Tao et al., 2022b), and specifically seq2seq models has been relatively much lower. Recent work like, EdgeFormer, (Ge and Wei, 2022), LLM.int8() (Dettmers et al., 2022) have recently demonstrated the generation of seq2seq quantized models which provide a high-compression ratios and competitive performances though this work has also been done with much higher resource languages.

Distillation Work within distillation is replete, even for the multilingual-type of models that we focus on. Work like Kaliamoorthi et al. (2021); Jiao et al. (2021); Yang et al. (2022) represent the major body of work in multi-lingual distillation - that is also centered across the encoder-only space. Relatively lesser work has been done in the space of mutli-lingual distillation (Soltan et al., 2021; Mukherjee et al., 2021) of seq2seq models and even though work like Zhang et al. (2020); He et al. (2019) extends this analysis to relatively low-resource languages, they rely on the use of monolingual data for the target language, a luxury that we cannot afford for half of the languages in our language set.

Note that since both processes are orthogonal, their conjunctive use has also been explored - Tao et al. (2022a) for instance, get competitive results by applying token level contrastive distillation and module-wise dynamic scaling while quantizing generative models. Note that we made the conscious decision of excluding pruning from our analysis because while it is known to demonstrate very effective parameter reduction, it is generally not as aggressive in it's memory footprint reduction as much as quantization and distillation (Behnke and Heafield, 2020; Mohammadshahi et al., 2022). As we'll discuss further in section 7, size-reduction was an implicit focus of this work that is one of the most fundamental bottlenecks of community deployment A.4.

7 Discussion

While this work explicitly focuses on only the performance comparison between distillation and post-training quantization, it's efficacy can also be viewed in demonstrating the development of lightweight, machine translation models for extremely low-resource languages. This is a very critical outcome as Performance-oriented Machine translation (MT) models for low-resource languages are often not suited for the immediate consumption of the community. The access bottleneck introduced by these bloated models, can especially affect those communities which haven't traditionally enjoyed access to a digital ecosystem, often widening the gap between those who can and cannot access these tools. Towards this direction, the exploration of compression strategies for these models - especially when tied to end-user centric NLP services such as translation is imperative. In this work, the size of all models being evaluated after compression was less than 400MB - the quantized models are at least 3x lighter the size of the native HM and the distilled models give even more impressive gains of upto 8x smaller than their uncompressed counterparts. This size reduction, coupled with the increased speed of inference associated with this reduction in most cases can enable a suite of accessible translation models for these languages[5]. This establishes a very promising potential in achieving deployment-constraint aware models: For instance, in areas where users do not enjoy a sustained access to the internet - these lightweight models may be adapted to operate on edge in an offline fashion.

8 Conclusion and Future Work

In this work we established that hard-distillation is sensitive to several priors which makes it a brittle mechanism of compression, especially for languages with extremely low-resources. In relative comparison, post-training quantizaton provides a competitive, stable and cost-effective compression mechanism that works effectively for extremely low-resource languages as well. Moving forward, we wish to explore the effect of using additional data (augmented or natively available) on the compressed variants of these models and extend distillation's analysis to utilizing logit distributions of the teacher (soft-distillation). Having observed the poor confidence measures of the transformer - and it's relatively random distributions we expect to get more interpretable evidence towards the suitability of these models for soft distillation through such an analysis.

[5]A more detailed description of the sizes of these models and the associated inference patterns is provided in the Appendix A.4

Acknowledgements

We sincerely thank the reviewers for their detailed feedback on the work which greatly helped us improve the quality of this work. Additionally, we thank Indian Institute of Technology, Kharagpur for giving us access to the Mundari data. Finally, we also thank Anurag Shukla for helping formalize the quantization pipeline and Kabir Ahuja, Sumanth Doddapaneni and Naman Jain for all the helpful discussions.

References

Àlex R Atrio and Andrei Popescu-Belis. 2022. Small batch sizes improve training of low-resource neural mt. arXiv preprint arXiv:2203.10579.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, et al. 2022. Building machine translation systems for the next thousand languages. arXiv preprint arXiv:2205.03983.

Maximiliana Behnke, Nikolay Bogoychev, Alham Fikri Aji, Kenneth Heafield, Graeme Nail, Qianqian Zhu, Svetlana Tchistiakova, Jelmer van der Linde, Pinzhen Chen, Sidharth Kashyap, and Roman Grundkiewicz. 2021. Efficient machine translation with model pruning and quantization. In Proceedings of the Sixth Conference on Machine Translation, pages 775–780, Online. Association for Computational Linguistics.

Maximiliana Behnke and Kenneth Heafield. 2020. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2664–2674, Online. Association for Computational Linguistics.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. arXiv preprint arXiv:2109.12948.

Tommaso Caselli, Roberto Cibin, Costanza Conforti, Enrique Encinas, and Maurizio Teli. 2021. Guiding principles for participatory design-inspired natural language processing. In Proceedings of the 1st Workshop on NLP for Positive Impact, pages 27–35, Online. Association for Computational Linguistics.

Swara CGNET. 2019. Hindi-gondi parallel corpus. https://arxiv.org/abs/2004.10270.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. arXiv preprint arXiv:2208.07339.

Harshita Diddee, Kalika Bali, Monojit Choudhury, and Namrata Mukhija. 2022. The six conundrums of building and deploying language technologies for social good. In ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS), COMPASS '22, page 12–19, New York, NY, USA. Association for Computing Machinery.

Harshita Diddee and Bhrigu Kansra. 2020. Crosspriv: User privacy preservation model for cross-silo federated software. In 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), pages 1370–1372.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. arXiv preprint arXiv:1808.09381.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of NLP leaderboards. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4846–4853, Online. Association for Computational Linguistics.

Isaac Feldman and Rolando Coto-Solano. 2020. Neural machine translation models with back-translation for the extremely low-resource indigenous language Bribri. In Proceedings of the 28th International Conference on Computational Linguistics, pages 3965–3976, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. Transactions of the Association for Computational Linguistics, 8:539–555.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. 2021. Compressing Large-Scale Transformer-Based Models: A Case Study on BERT. Transactions of the Association for Computational Linguistics, 9:1061–1080.

Tao Ge and Furu Wei. 2022. Edgeformer: A parameter-efficient transformer for on-device seq2seq generation. arXiv preprint arXiv:2202.07959.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, 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.

Tianyu He, Jiale Chen, Xu Tan, and Tao Qin. 2019. Language graph distillation for low-resource machine translation. arXiv preprint arXiv:1908.06258.

Tao Huang and Hong Chen. 2021. Improving privacy guarantee and efficiency of Latent Dirichlet Allocation model training under differential privacy. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 143–152, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, and Sanjeev Arora. 2020. TextHide: Tackling data privacy in language understanding tasks. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1368–1382, Online. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2021. Lightmbert: A simple yet effective method for multilingual bert distillation. arXiv preprint arXiv:2103.06418.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6282–6293, Online. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in c++. arXiv preprint arXiv:1805.12096.

Prabhu Kaliamoorthi, Aditya Siddhant, Edward Li, and Melvin Johnson. 2021. Distilling large language models into tiny and effective students using pqrnn. arXiv preprint arXiv:2101.08890.

Yash Khemchandani, Sarvesh Mehtani, Vaidehi Patil, Abhijeet Awasthi, Partha Talukdar, and Sunita Sarawagi. 2021. Exploiting language relatedness for low web-resource language model adaptation: An indic languages study. arXiv preprint arXiv:2106.03958.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. In International conference on machine learning, pages 5506–5518. PMLR.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. 2020. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In International Conference on Machine Learning, pages 5958–5968. PMLR.

Manuel Mager, Arturo Oncevay, Abteen Ebrahimi, John Ortega, Annette Rios, Angela Fan, Ximena Gutierrez-Vasques, Luis Chiruzzo, Gustavo Giménez-Lugo, Ricardo Ramos, Ivan Vladimir Meza Ruiz, Rolando Coto-Solano, Alexis Palmer, Elisabeth Mager-Hois, Vishrav Chaudhary, Graham Neubig, Ngoc Thang Vu, and Katharina Kann. 2021. Findings of the AmericasNLP 2021 shared task on open machine translation for indigenous languages of the Americas. In Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas, pages 202–217, Online. Association for Computational Linguistics.

Nina Markl. 2022. Mind the data gap(s): Investigating power in speech and language datasets. In Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, pages 1–12, Dublin, Ireland. Association for Computational Linguistics.

Idriss Mghabbar and Pirashanth Ratnamogan. 2020. Building a multi-domain neural machine translation model using knowledge distillation. arXiv preprint arXiv:2004.07324.

Alireza Mohammadshahi, Vassilina Nikoulina, Alexandre Berard, Caroline Brun, James Henderson, and Laurent Besacier. 2022. What do compressed multilingual machine translation models forget? arXiv preprint arXiv:2205.10828.

Subhabrata Mukherjee, Ahmed Hassan Awadallah, and Jianfeng Gao. 2021. Xtremedistiltransformers: Task transfer for task-agnostic distillation. arXiv preprint arXiv:2106.04563.

Toshiaki Nakazawa, Hideki Nakayama, Isao Goto, Hideya Mino, Chenchen Ding, Raj Dabre, Anoop Kunchukuttan, Shohei Higashiyama, Hiroshi Manabe, Win Pa Pa, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, Katsuhito Sudoh, Sadao Kurohashi, and Pushpak Bhattacharyya, editors. 2021. Proceedings of the 8th Workshop on Asian Translation (WAT2021). Association for Computational Linguistics, Online.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 311–318.

Maja Popović. 2016. chrF deconstructed: beta parameters and n-gram weights. In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pages 499–504, Berlin, Germany. Association for Computational Linguistics.

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

Akshai Ramesh, Venkatesh Balavadhani Parthasarathy, Rejwanul Haque, and Andy Way. 2021. Comparing statistical and neural machine translation performance on hindi-to-tamil and english-to-tamil. Digital, 1(2):86–102.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. Transactions of the Association for Computational Linguistics, 10:145–162.

Surangika Ranathunga and Nisansa de Silva. 2022. Some languages are more equal than others: Probing deeper into the linguistic disparity in the nlp world. arXiv preprint arXiv:2210.08523.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In International Conference on Machine Learning, pages 4596–4604. PMLR.

Saleh Soltan, Haidar Khan, and Wael Hamza. 2021. Limitations of knowledge distillation for zero-shot transfer learning. In Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing, pages 22–31, Virtual. Association for Computational Linguistics.

Mukund Srinath, Shomir Wilson, and C Lee Giles. 2021. Privacy at scale: Introducing the PrivaSeer corpus of web privacy policies. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6829–6839, Online. Association for Computational Linguistics.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022a. Compression of generative pre-trained language models via quantization. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4821–4836, Dublin, Ireland. Association for Computational Linguistics.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022b. Compression of generative pre-trained language models via quantization. arXiv preprint arXiv:2203.10705.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In Proceedings of the Fifth Conference on Machine Translation, pages 1174–1182, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems, 30.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. arXiv preprint arXiv:2203.09435.

Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. 2020. Integer quantization for deep learning inference: Principles and empirical evaluation. arXiv preprint arXiv:2004.09602.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7361–7373, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 483–498, Online. Association for Computational Linguistics.

Ziqing Yang, Yiming Cui, Zhigang Chen, and Shijin Wang. 2022. Cross-lingual text classification with multilingual distillation and zero-shot-aware training. arXiv preprint arXiv:2202.13654.

Ali Hadi Zadeh, Isak Edo, Omar Mohamed Awad, and Andreas Moshovos. 2020. Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference. In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pages 811–824.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. In 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS), pages 36–39.

Shiyue Zhang, Ben Frey, and Mohit Bansal. 2022. How can NLP help revitalize endangered languages? a case study and roadmap for the Cherokee language. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1529–1541, Dublin, Ireland. Association for Computational Linguistics.

Xinlu Zhang, Xiao Li, Yating Yang, and Rui Dong. 2020. Improving low-resource neural machine translation with teacher-free knowledge distillation. IEEE Access, 8:206638–206645.

A Appendix

A.1 Details of Data Sources

For all the languages in Table 1 we now describe the training and evaluation corpora used. Note that for languages like Assamesse, Odia, Punjabi, etc. we could have accessed a monolingual corpus to supplement our training as well but since we wouldn't have been able to leverage data at a similar scale and quality for the entire language set, we abstained from using methods that leveraged monolingual corpora in these languages.

Bribri Training data from Feldman and Coto-Solano (2020) containing about 7K parallel sentences. Test data from Mager et al. (2021) with 1003 sentences.

Wixarica Training data from Feldman and Coto-Solano (2020) containing about 8k parallel sentences. Test data from Mager et al. (2021) with 1K sentences.

Mundari We requested Indian Institute of Kharagpur for Data on Mundari. This corpus contained 10K parallel sentences. We partition train and test sets from this and generate a test set of 980 sentences [6]

Gondi Data obtained from CGNET (2019) containing 26K sentences. We partition train and test sets from this and generate a test set of 730 sentences6.

Assamesse Train data obtained from Ramesh et al. (2022) containing 0.14 parallel sentences. Test data from (Goyal et al., 2022) containing 1012 sentences

[6] To avoid any test-set leaks, we deduplicate the data by removing tuples (S^i, T^i) where S^i is the i^{th} sentence in the source language and T^i is $i^{th}$the sentence in the target language, between the train and the test set.

Odia Train data obtained from Ramesh et al. (2022) containing 1M parallel sentences. Test set from WAT2021 (Nakazawa et al., 2021) containing 2390 sentences

Punjabi Train data obtained from Ramesh et al. (2022) containing 2.42M parallel sentences. Test set from WAT2021 (Nakazawa et al., 2021) containing 2390 sentences

Gujarati Train data obtained from Ramesh et al. (2022) containing 3.05M parallel sentences. Test set from WAT2021 (Nakazawa et al., 2021) containing 2390 sentences

A.2 Evaluating Continued Pretaining with Synthetically Augmented or Lexicon Adapted Monolingual Data for improving the HM

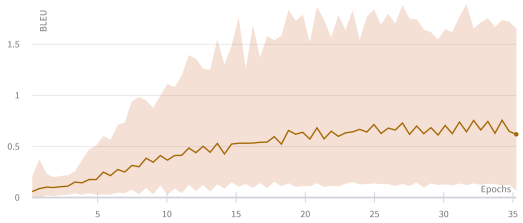
The use of continual pretraining with monolingual data has been shown to be very useful in improving the transfer for low-resource languages. In our cases, our lowest resource languages, i.e, Bribri, Wixarica, Gondi and Mundari, did not have any monolingual data available natively so we explored the augmentation of the same using lexicons (Wang et al., 2022). We also generated forward translated data using the HM that we developed to fuse with the lexicon-adapted data. For continued pretraining we use a fixed learning rate of 0.001. Results of our experiments are logged in Table 5.We use the following notations to report our results *GMD- Gold Monolingual Data, LA- Lexicon Adapted Monolingual Data, KDD- Knowledge Distilled Monolingual Data* where GMD indicates the target-side monolingual data available within the parallel corpus of the language, KDD indicates the forward-translated data that we generate via our best-performing model for Gondi i.e., mt5-base. We generated 100K labels using mt5-base teacher, and also experimented adding 100K sentences from a weaker teacher, i.e., mt5-small in hopes of leveraging a more diverse class of labels to train the student on.

We did observe a small gain in performance upon the addition of LA data during pretraining - though the post-quantization performance and the distilled model' significant performance degradation called for a deeper investigation on the effects of continued pretraining for this language.

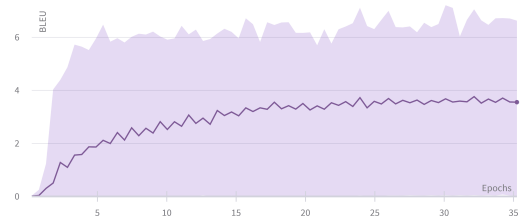
A.3 Hyperparameter Trial Configurations

We ran Hyperparameter sweeps with the configurations specified in Table 5.

Note that in congruence with the observations of subsection 4.6, we also provide the min-max range of performance for Gondi and Bribri in Figure 6.



(a) Min/Max Range of Bribri's Sweep



(b) Min/Max Range of Gondi's Sweep

Figure 6: Variation of performance across languages

As can be observed, for a set of hyperparameters, at least one of which is optimal for some other language in the set, both languages fail to converge. Similarly, in extension to subsection 4.4, we also checked if for the same hyperparameter set, the variation in student architecture produced significant performance variations.

The results demonstrated in Figure 7 did not show any significant variation except for the case of Gondi, i.e., altering the student architecture - while keeping all other priors the same: adversely affected the performance in that one case.

A.4 Comparing Size-Reduction Affinity of Quantization and Distillation

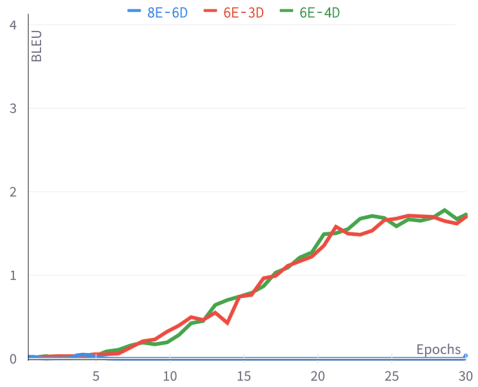
This exploration is extremely useful as the size of a model significantly impacts several factors associated with the consumption of any service, impacting it's adoption by community members through several ways including *(a) Accessibility on Edge:* Since mobile devices are constrained in their RAM and Memory Usage - users with edge devices of low-capabilities are naturally inhibited to is services that drain their device's resources. *Inadequate Connectivity Requirement for Inference, One-time download and Service Updates:* Users

Model	Data	spBLEU	S(M) (in MB)
Transformer	$26.2k$	1.4	240
mT5-small	$61.9k$	12.7	1200
mT5-small	$26.2k$	14.3	1200
mT5-base	$26.2k$	15.6	2100
mBART	$26.2k$	13	2280
mT5-small: CPT {GMD }	$26.2k^{mono}$	14.9	1200
mT5-small: CPT {LA }	$200k^{mono}$	**14.9**	1200
mT5-small: CPT {LA }	$200k^{mono}$	**10.8**	400
mT5-small: CPT {KDD }	$143k^{mono}$	15.2	1200
mT5-small: CPT {GMD + LA + KDD }	$26.2k + 343k^{mono}$	14.7	1200
mT5-small: Quantizing M1	$26.2k$	13.8	400
Quantizing CPT Model {Best mT5-small }	$26.2k$	10.2	400
Transformer + KD	$26.2k + 240k$	10.1	185

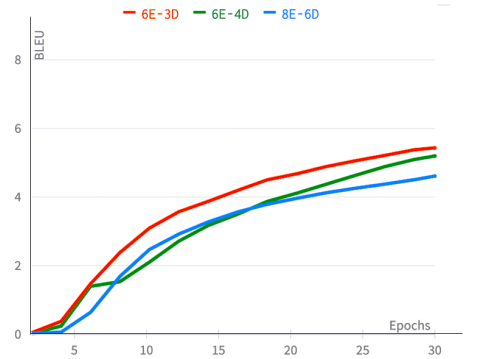
Table 4: Gondi: Use of Lexicon Adaptation, Continued Pretraining and Mixed-training with Lexicon Adapted and Forward Translated Monolingual Data.

Hyperparameter	Candidate Values
Train batch size	32, 64
Epochs	10, 30, 60
Method	grid
Metric	BLEU
Gradient Accumulation	2, 4
Label Smoothing	0, 0.1
Learning Rate	5{e-5,e-5,e-6}
Warmup Steps	500, 1000

Table 5: Candidate values of hyperparameters: Sweep for finding the optimal hyperparameter set for Distillation



(a) Variation in BLEU with change in student architecture for Assamesse



(b) Variation in BLEU with change in student architecture for Gondi

Figure 7: In the legend E and D refers to Encoders and Decoders respectively

may often avoid downloading apps that seem too large, particularly in emerging markets where devices connect to often-spotty 2G and 3G networks or work on pay-by-the-byte plans [7]. *Large Rendering Time:* Finally, a bloated size may often be associated with a larger rendering response period which might hinder the usability experience of a user engaging with the MT service.

Note on Inference Times In theory, compression through both distillation and quantization is expected to be conducive to faster inference for the models: The distilled models are not bounded to use a pretrained embedding and hence can gain in inference by using smaller, target-language specific embeddings. The quantized models can also benefit due to the reduced precision in which the inference operations are carried out, though this optimization is heavily dependent on if the hardware running the model can leverage these operations in

[7] https://developer.android.com/topic/performance/reduce-apk-size

Language	Native S(HM)	Compressed S(Q,D)
Bribri	1228	(400, 153)
Wixarica	1228	(400, 153)
Gondi	1228	(400, 153)
Mundari	1228	(400, 153)
Assamesse	1228	(400, 189)
Odia	1228	(400, 189)
Punjabi	232	(75, 189)
Gujarati	232	(75, 189)

Table 6: Sizes of the Uncompressed and Compressed Variants for all languages - Q and D indicate the compressed sizes of the Quantized and the Distilled Models respectively. All sizes are in MB.

their expected precision (Bondarenko et al., 2021). Especially in the case of quantization, the scope of this analysis would be quite vast, which is why we also excluded it from our current analysis.