The Low-Resource Double Bind: An Empirical Study of Pruning for Low-Resource Machine Translation

Orevaoghene Ahia
Masakhane NLP
oreva.ahia@gmail.com

Julia Kreutzer
Google Research
Masakhane NLP
jkreutzer@google.com

Sara Hooker
Google Research, Brain
shooker@google.com

Abstract

A "bigger is better" explosion in the number of parameters in deep neural networks has made it increasingly challenging to make state-of-the-art networks accessible in compute-restricted environments. Compression techniques have taken on renewed importance as a way to bridge the gap. However, evaluation of the trade-offs incurred by popular compression techniques has been centered on high-resource datasets. In this work, we instead consider the impact of compression in a data-limited regime. We introduce the term *low-resource double bind* to refer to the co-occurrence of data limitations and compute resource constraints. This is a common setting for NLP for low-resource languages, yet the trade-offs in performance are poorly studied.

Our work offers surprising insights into the relationship between capacity and generalization in data-limited regimes for the task of machine translation. Our experiments on magnitude pruning for translations from English into Yoruba, Hausa, Igbo and German show that in low-resource regimes, sparsity preserves performance on frequent sentences but has a disparate impact on infrequent ones. However, it improves robustness to out-of-distribution shifts, especially for datasets that are very distinct from the training distribution. Our findings suggest that sparsity can play a beneficial role at curbing memorization of low frequency attributes, and therefore offers a promising solution to the low-resource double bind.

1 Introduction

Over the years, the size of language models have grown exponentially (Amodei et al., 2018; Thompson et al., 2020; Bender et al., 2021). Additional parameters have improved quality on a variety of downstream NLP tasks, but drive up the cost of training (Horowitz, 2014; Strubell et al., 2019; Patterson et al., 2021) and increase the latency and memory footprint at inference time (Warden and Situnayake, 2019; Samala et al., 2018).
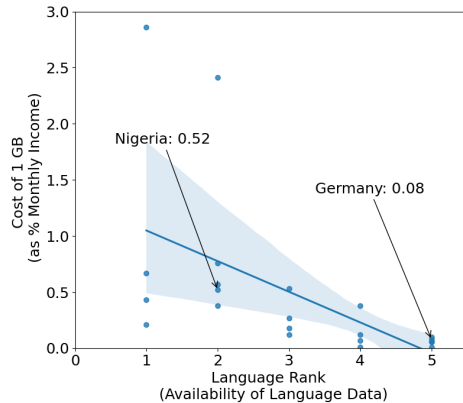

Figure 1: Cost of mobile data by country per language rank according to the taxonomy by Joshi et al. (2020).

Extending state-of-the-art language models to low-resource languages requires addressing what we term the *low-resource double bind*. Low-resourcedness goes beyond mere data availability and reflects systemic issues in society (Martinus and Abbott, 2019; Nekoto et al., 2020). Classifications of languages with respect to "resourcedness" have focused on the relative availability of data (Zoph et al., 2016; Joshi et al., 2020), and the concentration of NLP researchers from these regions or the over-fitting of model design around a small set of high resource languages (Cieri et al., 2016; Nekoto et al., 2020).

Less well documented and explored is the over-indexing of low-resource languages in ecosystems which simultaneously present severe constraints of computational resource. In Fig. 1 we plot 22 languages grouped by the availability of labelled and unlabelled data as proposed by Joshi et al. (2020) against the cost of 1 GB of data as a percentage of monthly income. Each language is mapped to the country with the most speakers. The cost of data is a valuable proxy for the cost of access to technology in an ecosystem (Oughton, 2021). Here, this visibly co-occurs with the limitations in available data for different languages.

In computationally constrained environments, access to machine learning technology depends upon optimizing jointly for both model performance and compactness. Pruning and quantization are widely applied techniques for compressing deep neural networks prior to deployment, as compressed models require less memory, energy consumption and have lower inference latency (Esteva et al., 2017; Lane and Warden, 2018; Sun et al., 2020). To-date, evaluating the merits and trade-offs incurred by compression have overwhelmingly centered on settings where the data is relatively abundant (Gale et al., 2019; Li et al., 2020a; Hou et al., 2020; Chen et al., 2021; Bai et al., 2020; ab Tessera et al., 2021).

In this work, we *instead* ask how these design choices trade-off with performance in data-limited regimes typical of low resource languages. We conduct large scale experiments on Neural Machine Translation (NMT) models trained to translate between English and three low resource African languages (Yoruba, Igbo and Hausa) and one high resourced language (German). We compare performance across models independently trained to very different levels of sparsity — ranging from 50 % to 98 % — and evaluate performance on the original distribution, in addition to establishing sensitivity to distribution shift across multiple corpora.

Recent work restricted to the computer vision domain has found that sparse models with comparable top-line performance metrics diverge considerably in behavior on the long-tail of the distribution and are sensitive to distribution shifts (Hooker et al., 2020a; Liebenwein et al., 2021). Here, we rigorously characterize the impact of sparsity on learned decision boundaries in NMT. In addition to held-out set BLEU, we measure sub-group performance on sentences grouped by prototypicality and study generalization properties over test corpora with different out-of-vocabulary ratios. We also evaluate whether humans prefer translations from sparse or dense models.

Our contributions can be enumerated as follows:

1. We introduce the term *low-resource double-bind* and develop an extensive experimental framework to understand the impact of compression in a data-limited regime across 4 languages and 5 different data sets.

2. We find that models are *tolerant of high levels of sparsity* while retaining BLEU performance and also human-judged translation quality.

This holds until extremely high levels of sparsity (95%–99% of all weights removed) where a severe decline in BLEU is notable.

3. There is a more pronounced degradation when evaluation includes less frequent input patterns. On closer investigation, we find that *sparsity disproportionately degrades performance on the long-tail of the data distribution.*

4. Curbing memorization of the long-tail can provide unexpected benefits. In a data-limited regime, we find that *sparsity benefits generalization to out-of-distribution corpora.*

Implications of Our Work Understanding the impact of compression on low-resource languages is key to making technology accessible and inclusive. Our work suggests that compression in these settings alters generalization in ways that can be beneficial and go beyond merely fulfilling deployment constraints. A challenge in low-resource NLP is that the existing publicly available corpora often come from very specific domains, such as missionary websites or translations of religious texts. These sources do not adequately reflect the reality of the potential applications of NLP technologies, and are rarely sufficient for deployment (Öktem et al., 2020; Anastasopoulos et al., 2020; Öktem et al., 2021). Thus, a task of great interest is establishing what model design choices can lead to generalization properties that extend beyond the immediate task at hand. Our work suggests that sparsity can play an important role in aiding generalization by curbing the memorization of rare long-tail instances.

2 Methodology

Addressing the low-resource double bind requires a careful setup of experiments to reflect the realities of low-resource translation. In particular, we want to control the effects of (1) network sparsity, (2) training data size, (3) target language, and (4) domain shifts.

In this work we focus on pruning, a widely favored compression technique due to remarkably high levels of compression that can be achieved while retaining top-line performance (Gale et al., 2019). Pruning typically involves three separate stages: 1) training a dense model, 2) progressively removing a subset of weights estimated to be unimportant, and 3) continuing to train the smaller sparse network for a certain number of steps to

recoup performance (Reed, 1993; Blalock et al., 2020). Pruning is the subject of considerable research and numerous techniques have been proposed, which differ in how weights are identified for removal and the schedule for introducing sparsity/allowing recovery (Cun et al., 1990; Hassibi et al., 1993a; Ström, 1997; Louizos et al., 2017; See et al., 2016; Evci et al., 2019; Narang et al., 2017). The development of specialized software kernels has enabled the acceleration of sparse networks on traditional hardware (Gale et al., 2020; Elsen et al., 2019; Zhu et al., 2019) with new generations of hardware directly facilitating sparse training (Zhu et al., 2019).

State of art pruning techniques can achieve a far higher level of compression and performance than simply using a smaller dense network (Zhu and Gupta, 2017; Li et al., 2020b). In our setting, a 90% sparse base transformer greatly outperforms a tiny dense one across all the languages despite having a fraction of the parameters (14M vs 4.6M) (Appendix Table 8).

2.1 Magnitude Pruning

We use magnitude pruning (Zhu and Gupta, 2017) to introduce sparsity across all experiment variants. It consistently achieves comparable or better results than competing state of art approaches on large scale benchmarks of computer vision and language models (Gale et al., 2019) and is widely used in practice due to the ease of implementation. Magnitude pruning estimates weight importance as the absolute weight value, and removes the weights with lowest magnitude according to a pre-specified schedule which determines the interval of training steps and frequency between begin and end step across which sparsity is introduced.

Magnitude pruning allows for the pre-specification of desired sparsity such that we can train models from random initialization to precise levels of end sparsity. We carry out extensive experiments and train networks independently for each language to end sparsity of 0–98 where 98% designates a network with 98% of the weights removed by the end of training. 0% is a dense network (no weights removed).

2.2 Languages

We validate the effectiveness of magnitude-based pruning method in NMT models trained to translate from English into German (`de`), Yoruba (`yo`), Igbo (`ig`) and Hausa (`ha`). While German as a

| | Training | Distribution Shift Test | | | | | |
	JW300	Gnome	Ubuntu	Flores	ParaCrawl	Tanzil	Tatoeba
`de`	1.9M	5,963	11,161	1012	2,000	2,000	10,145
`yo`	414.0k	1,467	120	1012	-	-	-
`ig`	414.9k	3,173	608	1012	2,000	-	-
`ha`	211.9k	998	219	1012	2,000	2,000	-

Table 1: Number of sentences in each parallel corpora we evaluate. For ParaCrawl and Tanzil, we sample 2000 sentences from the full dataset.

high-resource language serves as a point of comparison to previous works, Yoruba, Igbo and Hausa represent three of the highest-resource African languages with (near-)sufficient resources for reliable MT experimentation, i.e. multiple publicly-available parallel corpora. Joshi et al. (2020) classify Yoruba and Hausa as "Hopeful" in terms of available NLP resources and research, whereas Igbo is slightly lower-resourced and classified as "Scraping-by". All constitute important test beds for developing technologies that improve treatment of low-resource technologies, since they each have more than 50 million native speakers. Yoruba and Igbo belong to the Niger-Congo language family and use diacritics that pose challenges for text-based NLP (Orife, 2018; Dossou and Emezue, 2021). Hausa is a Chadic language which is part of the Afroasiatic phylum. It features complex pluralization and agglutination.

2.3 Training and Test Data

JW300 Training data for all languages is obtained from the JW300 parallel corpus (Agić and Vulić, 2019), since it is the largest source of data that covers all languages we evaluate. It comprises more than 300 languages of which 101 are African, and is collected from religious magazines by Jehovah's Witnesses (JW) published on `jw.org`.

Pre-processing Parallel sentences are tokenized and encoded using BPE (Sennrich et al., 2016), resulting in a shared vocabulary of 4096 tokens. Sentences are batched together with a maximum sequence length of 64. For each training batch, the approximate number of source/target tokens is 2048. We compute detokenized and case-sensitive BLEU using a helper script in tensor2tensor (Vaswani et al., 2018) equivalent to SacreBLEU (Post, 2018).

Full vs limited data regime For our experiments, we train on these datasets in two settings: First, with all data available for each of the languages, sizes listed in Table 1. In this setting, the dataset sizes range from 212k for Hausa to 1.9M
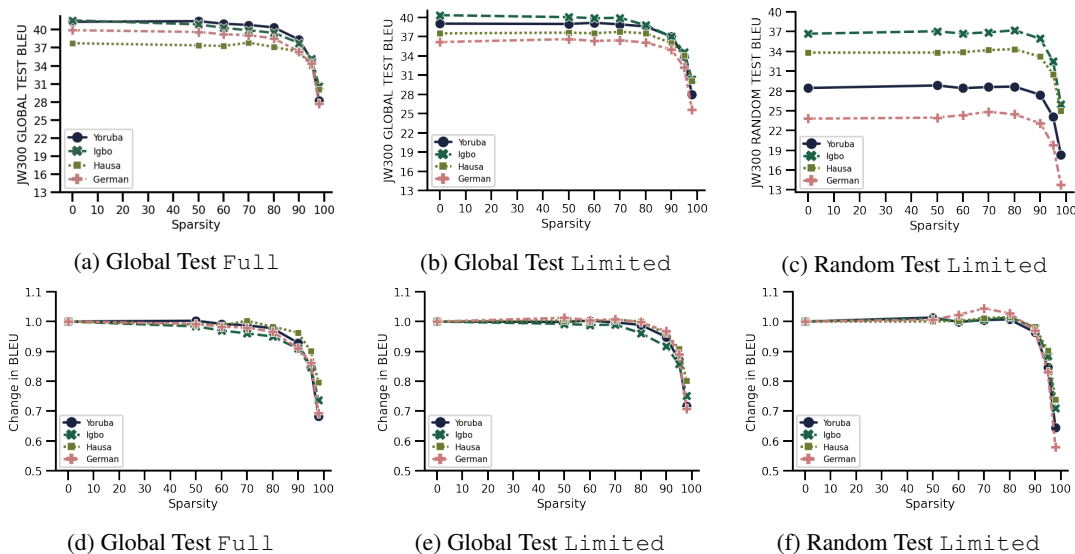
Figure 2: Impact of pruning on BLEU performance across languages, sparsity levels and training data regimes. We evaluate test set performance on both a `Global` test set designed around common phrases to allow comparability between data corpus, and a `Random` test set with sentences sampled at random. **Top row:** Absolute change to BLEU and by test set and sample size **Bottom row:** Change in BLEU relative to the dense (0% sparse) model.

for German. Our second setting holds constant the amount of data available by sampling a uniform number of sentence pairs for each language. We randomly sample 200k sentences from the train set of each language, limited by the smallest corpus Hausa which consists of approximately 210k sentences. We refer to these settings in experiment discussion as `Full` and `Limited`.

Validation & testing   The need for multiple test sets to capture performance on a variety of downstream conditions has already been recommended by recent work (Søgaard et al., 2021; Lazaridou et al., 2021). The JW300 test sets were constructed and released by Nekoto et al. (2020) to contain the most frequent sentences in the JW300 corpus across African languages and were filtered from the training corpus. This construction ensures that test sets across languages contain similar content, which leads to increased comparability. However, this cross-lingual selection may introduces a bias towards frequent sentences, and under-represents language-specific outliers.

Only measuring performance on frequent sentences across languages may be a particular concern in evaluating the impact of sparse models, as prior work has shown that the introduction of sparsity disproportionately impacts the long-tail of the data distribution (Hooker et al., 2020a,b). To capture possible disparate impact on the long-tail, we

also sample at random from the remainder of the data to craft a secondary test set (as has been done for validation). In the results section, we refer to the Nekoto et al. (2020) test data as the `Global` test set and random sample as the `Random` test set. A comparison of differences in performance between `Global` and `Random` test sets provides insights into how sparsity impacts generalization performance on text which is common relative to a more typical Zipf distribution with long-tail features (Zipf, 1999).

2.4   Sensitivity to Distribution Shift

We select corpora which differ from the training distribution in both domain (ranging from everyday sentences to technical documentation), sentence length and OOV rate (ranging from 2.68% to 20.42%). Given these large deviations in statistics from the JW300 training corpus, our expectation is *not* that the model preserves performance but rather to understand the sensitivity of sparse models to distribution shift *relative* to dense models.

Our selection of corpora is also guided by the size of public data sets that cover Yoruba, Hausa, Igbo and German. When the test set is small, reliability in BLEU scores between models and inferred conclusions may be compromised (Card et al., 2020). To estimate the impact that limitation in test size can have on our results, we simulate the variability of BLEU under different amounts
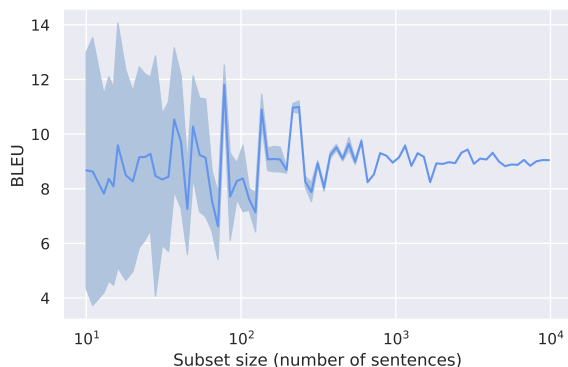
Figure 3: Mean BLEU scores (shaded: ± standard variation) for the dense en-de models on subsets of the Tatoeba data.

of test data in Figure 3. As can be seen, a sample size of at least 100 reasonably reduces variance in BLEU, so we only investigate out-of-distribution sensitivity with datasets of at least that size.

The domains of the datasets can be characterized below. Statistics for corpus sizes are given in Table 1, and out-of-vocabulary rates (OOV) and average source lengths in Table 3.

2.5 Datasets evaluated

The domains of each dataset considered are characterized below. Additionally, we include statistics for 1) corpus size in Table 3, and 2) out-of-vocabulary rates (OOV) and average source lengths in Table 1:

- **Gnome** is a dataset in the technical domain that contains 187 languages pairs derived from the translation of GNOME documentation.[1] The size of test sets for this corpus ranges between 998 (Hausa) and 5,963 (German). "Sentences" are often entities or phrases, with an average length of only 6-9 tokens.

- **Ubuntu** is a dataset in the technical domain. It consists of 42 languages pairs generated from translating localization files of the Ubuntu OS.[2] The size of test sets for this corpus ranges between 120 (Yoruba) and 11,161 (German), and it shows similar length statistics to GNOME.

- **Tanzil** is a religious dataset with 42 language pairs. It is a collection of Quran translations compiled by the Tanzil project.[3] We sample

2000 sentences for both German and Hausa, which have an average length of 23 tokens, being slightly longer than the average JW300 training sentence.

- **ParaCrawl** is a dataset obtained from mining the web for parallel sentences (Bañón et al., 2020). v8 covers 41 mostly European languages, but a pre-release of Igbo and Hausa allowed evaluation here. [4] The crawled websites for Hausa and Igbo are largely religious but some also publish news. We sample 2000 sentences for Hausa, Igbo and German with an average length of 22 tokens.

- **Tatoeba** is a crowdsourced dataset of short sentences concerning every day life translated by users of `https://tatoeba.org/`. We only report Tatoeba results for German as this is the only corpus with more than 100 sentences. Tatoeba sentences have similar length to Gnome and Ubuntu, but are full sentences.

- **Flores** is a multi-domain dataset containing professional translations of sentences extracted from English Wikipedia in 101 languages (Goyal et al., 2021). The size of test sets released for this corpus is 1012 sentences across all languages with similar length to Tanzil and Paracrawl.

Our choice of datasets is guided by a desire to capture datasets with different degrees of difference from the original training corpus. JW300 is a religious dataset, so one could expect more overlap with both **ParaCrawl** and **Tanzil** and far less with **Ubuntu** and **Gnome** which are both technical writing to document the use of a technology. We include **Flores** which covers a variety of different domains and finally **Tatoeba** for completeness, as a more general dataset consisting of everyday sentences.

2.6 Architecture and Training

We train transformer models (Vaswani et al., 2017) for each NMT task with a modified version of the tensor2tensor library (Vaswani et al., 2018) from (Gale et al., 2019). The transformer base model consists of 60M parameters, with 31% of the parameters in the attention layers, and 41%

---

[1] `https://www.gnome.org/`
[2] `https://ubuntu.com/`
[3] `https://tanzil.net/`

[4] `https://bit.ly/3f7WfVI`

	Training			Global test			Random test		
	Avg Len	Dense	90% Sparse	Avg Len	Dense	90% Sparse	Avg Len	Dense	90% Sparse
Low	33.01	62.04	24.93	23.58	31.41	29.45	32.73	22.35	23.08
Mid	18.26	80.09	26.82	14.03	35.37	34.68	17.53	23.99	23.58
High	8.99	78.91	28.58	9.86	48.56	48.05	9.09	25.47	24.86

Table 2: BLEU for different sets split according to sentence typicality for German, which is defined as average token log-frequencies in the training corpus (F_S in (Raunak et al., 2020)).

	JW300 Global		JW300 Random		Tanzil		Tatoeba		ParaCrawl		Gnome		Ubuntu		Flores	
	OOV	Len	OOV	Len	OOV	Len	OOV	Len	OOV	Len	OOV	Len	OOV	Len	OOV	Len
de	0.25	15.81	0.66	19.76	2.68	22.48	4.89	8.78	9.86	20.09	12.37	8.09	16.64	5.56		
ha	0.26	16.28	0.37	18.72	4.95	22.86	7.00	7.76	3.39	24.67	15.22	9.81	20.42	7.83	15.53	21.64
ig	0.30	15.98	0.50	18.58	-	-	12.10	6.89	6.95	20.91	14.19	6.99	13.99	6.81		
yo	0.24	15.98	0.56	18.77	-	-	9.05	5.69	-	-	16.66	6.36	13.55	6.46		

Table 3: Out-of-vocabulary rates (OOV, %) and average source lengths (Len) for different test set sources.

in the position wise feed-forward layers. Training hyperparameters are detailed in Appendix Section A.3. We release our code here https://github.com/orevaahia/mc4lrnmt.

Throughout training we introduce sparsity of levels percentages [0, 50, 60, 70, 80, 90, 95, 98] using magnitude pruning (Zhu and Gupta, 2017). All fully-connected layers and embeddings making up 99.87% of all of the parameters in the model are considered for sparsification. The tuning of pruning hyper-parameter is described in Appendix Section A.4.

2.7 Human Evaluation: Dense vs Sparse

We complement automatic BLEU evaluation with a human evaluation study to compare the translation quality of dense and sparse models. We elicit absolute ratings on a 6-point scale for 500 pairs of differing translations of the JW300 Global and Random test set on a crowd-sourcing platform.

3 Results

Sparsity BLEU trade-off In Figure 2, we can see that models are tolerant of moderate to high levels of sparsity (50% - 80%) while retaining BLEU performance relative to dense. Between 50% and 80% sparsity, any degradation is minimal as sparse performance remains at 95% or more of dense performance for all languages. Hausa even has a slight exception where pruning 70% or 80% of the model parameters performs on par with the baseline or even better. However, for both Global and Random test sets, there is a noticeably sharp degradation in BLEU when progressing to extremely high sparsity levels of 90% and beyond.

Long-tail test set Translation quality on the Global and Random test sets differs considerably. We control for data size by comparing on the same Limited datasets. Languages perform within a narrow band of comparable BLEU for Global, with the degradation in BLEU at higher levels of sparsity occurring at a similar rate across languages. In contrast, absolute BLEU scores on Random are noticeably lower at both dense and all sparsity levels, coupled with a far wider spread of BLEU between languages. This suggests that *a low data regime disproportionately impacts translation quality on the long-tail* and that the Random set is a more discriminative evaluation protocol. When we compare relative differences of sparse models to dense, we can see that relative to Global, there is sharper degradation in Random under high sparsity (90%+). However, with mid-level sparsity, the quality of the dense model is maintained or even slightly outperformed (German) on all test sets.

Learning prototypical instances is less sensitive to data size In Figure 2, it is noticeable that performance on the Global test set, does not vary noticeably between the Limited (2b) and Full (2a) training setting. This is surprising given the large difference in training corpus size for many of the languages (for German 1.9 M in Full vs 200,000 in Limited).

Additionally, even when restricting attention to the Full training setting, the ranking of absolute BLEU scores on the Global test set does not appear to be sensitive to the size of the training cor-
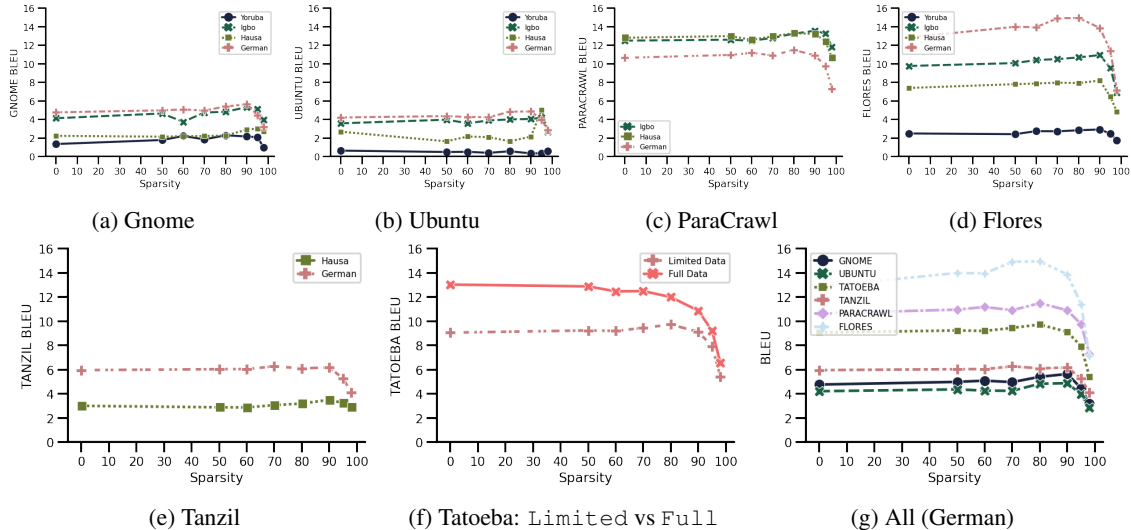
(a) Gnome	(b) Ubuntu	(c) ParaCrawl	(d) Flores

(e) Tanzil	(f) Tatoeba: Limited vs Full	(g) All (German)

Figure 4: Robustness to distribution shift at different levels of sparsity for models trained in a data-limited regime (`Limited`). For Tatoeba with only German we compare performance for a model trained on `Full` is added for comparison.

pus, as Igbo (414.9K) and Yoruba (414.0K) achieve nominally higher BLEU on `Global` than German (1.9M) despite having only a fifth of training data in the `Full` setup. This suggests that learning a representation for the most frequent patterns in the dataset does not require a substantial amount of data.

Data size for long-tail patterns In contrast, learning a good representation for the long-tail appears to be far more sensitive to the size of the training corpus. In Figure 4f, we compare the OOD performance on Tatoeba of the `Full` vs `Limited` model trained on German. Here, we are evaluating performance on a dataset with a much higher OOV ratio of 2.68%. Here, on a dataset with more rare instances and distribution shift, the amount of training data makes a larger difference. Across all levels of sparsity the model trained on `Full` generalizes better than the `Limited` model.

Do humans notice the difference between dense and sparse models? Human annotators rate test translations of the dense model and 90%-sparsity model as described in Section 2.7. Table 4 reports the average ratings (1-6 scale, the higher the better) for both types of models across languages for both in-domain test sets. The ratings reveal that there is no clear preference of dense over sparse model outputs across languages and sets. For German the sparse model scores 0.1 lower on average for both test sets. For Igbo and Hausa sparse scores slightly higher on the `Global` set, but this gain is lost on

`Random`. Hausa receives the nominally highest ratings on both test sets, but we note that raters might not be well calibrated across languages, and test sets are not completely identical. Nevertheless, the `Random` translations score consistently lower than the `Global` translations, indicating that the quality loss on long-tail examples was indeed noticeable. All in all, *the roughly 2-8% drop in BLEU that occurred through pruning at 90% is not negatively affecting human-judged translation quality in any of the studied languages*, which is a promising finding for the deployment of such models in practice. However, this evaluation is oblivious of effects like translation biases (Edunov et al., 2020) that could be caused by less memorization.

How sensitive are sparse models to distribution shift? Figure 4 shows the absolute change in BLEU when evaluating the `Limited` models on the out-of-distribution datasets.

We find that across dense and sparse models, degradation in performance is sensitive to OOV rates and difference in sentence lengths, with the most pronounced degradation on Tanzil (longer average sentence length), Ubuntu and Gnome (technical domains with far higher OOV rates of 12–20%). The transfer to ParaCrawl was the most successful across languages. For Flores, we don't see a uniform performance across all languages. Our results show that the transfer to all languages but Yoruba is almost similar to that of Paracrawl.

One trend that is consistent across languages and domains is an increase in quality around 70%–95%

sparsity (even more visible when plotting relative change in Figure 5 in the Appendix). As a result, 90% sparse models outperform their baselines across all languages under the `Limited` condition. This means that *increased sparsity during training with limited data has a beneficial influence for out-of-distribution generalization*. With larger amounts of training data—in the case of German (`Entire`) a factor of 10— however, this relative advantage is lost (Figure 4f). This finding is highly relevant for the reality of low-resource languages, where training data is limited and often available only for a narrow domain, so strong generalization is essential.

Does sparsity curb memorization? The results on OOD generalization are surprising, as it suggests that in a low-data regime *less capacity rather than more can aid generalization*. It is worth placing these results in the context of recent work in computer vision that has found that sparsity curbs memorization of the long-tail of the training distribution (Hooker et al., 2020a,b). Impeding memorization constrains model learning to the most general high level features, rather than atypical low-frequency features and noisy examples which typically require memorization and are less likely to generalize across other tasks (Brown et al., 2020; Feldman and Zhang, 2020). In the setting of low-resource languages, the training corpus is often highly restricted to a narrow domain such as religion. Here, it may be more likely the long-tail will contain specialized artefacts and noise that do not generalize.

To explore this further, in Table 2 we look at training and test performance grouped by sentence typicality for German (typicality measured as per (Raunak et al., 2020)). At training time, the dense model evidences clear overfitting and outperforms the sparse on low, mid and high typicality. The difference in relative performance on sentences of low typicality is striking (62.04 dense vs 24.93 sparse BLEU), confirming that capacity aids memorization of the long-tail. However, at test time the dense memorization of a specialized training set exhibits a negative knowledge transfer cost relative to sparse. On the `Random` test set, sparse in fact slightly outperforms relative to dense on low typicality. Both Table 2 and the OOD result in Figure 4 show that sparsity has an important role to play in data limited regimes at curbing memorization of rare artefacts that do not generalize.

	Global		Random	
	Dense	90% Sparse	Dense	90% Sparse
de	4.17	4.06	3.80	3.66
yo	3.66	3.66	3.51	3.51
ig	3.87	3.96	3.81	3.85
ha	4.77	4.85	4.53	4.53

Table 4: Average human ratings of 500 test set translations comparing **dense** and **90%-sparse** models (`Limited`) on `Global` and `Random` JW300 test sets.

4 Related Work

Compression techniques for NMT There has been recent works on compressing recurrent neural networks (RNN) and transformer networks for NMT (Gale et al., 2019; Narang et al., 2017; See et al., 2016; Zhang et al., 2017; Li et al., 2020b). With the exception of a proof-of-concept experiment with RNN on a small-scale English-Vietnamese translation task (See et al., 2016), all of the works above focus on compressing models trained on large data sets, and exclude African Languages.

To the best of our knowledge, our work is the first to apply pruning methods to train transformer NMT models on low-resourced data, and on African languages with different syntactic and morphological features distinct from English. Moreover, all of the above works rely solely on automatic evaluation metrics, and do not qualify translation quality using human annotation or sensitivity to different distribution shifts.

Optimized training for low-resource NMT Sennrich and Zhang (2019) find that hyperparameter tuning is essential for NMT on low-resource data, such as the depth or regularization of the network. Duh et al. (2020) highlight the importance of hyper-parameter tuning for NMT for Somali and Swahili. Fadaee et al. (2017); Sennrich and Zhang (2019) and Xu et al. (2020) explore tailored augmentation and curriculum learning strategies for data-limited regimes. Sennrich and Zhang (2019) additionally assume limitations to compute at training time when modeling Somali and Gujarati. In contrast, in our work, we consider the impact of resource constraints present at inference time/deployment of a model.

Transformer size for low-resource NMT More relevant to our work are works that have evaluated transformers of different sizes in the light of low-resource translation. Biljon et al. (2020) in-

vestigate the effect of transformer depth on low-resource translation for three South-African languages. Murray et al. (2019) study auto-size feed-forward and attention layers in transformers for low-resource translations of Hausa, Tigrinya, and Arabic, and find BLEU and efficiency improvements with smaller models. Tapo et al. (2020) succeed in training a smaller transformer model for Bambara with as few as 10k examples, but find only limited generalization under distribution shift (Tapo et al., 2021).

In contrast to these works, we study generalization at different levels of sparsity. Pruning is a more precise experimental framework to understand the relationship between capacity and generalization because we can exactly vary the sparsity in a range between 0% and 100% controlling for the same architecture. Pruning also achieves far higher levels of compression in terms of the number of parameters relative to substitutes evaluated in these works such as the tiny transformer. In our work, we also seek to not only measure the impact of capacity but also to better understand why counter-intuitively higher levels of sparsity aid generalization. Finally, our experiments are extensive relative to (Biljon et al., 2020; Tapo et al., 2020), both in terms of number of languages and variety of training and evaluation conditions. Furthermore, we are the first to report human evaluation on the effects of pruning for MT.

5 Future Work

Our work introduces the term *low-resource double-bind* and conducts extensive experiments to study the impact of pruning. In this setting, we are concerned with resource constraints present at deployment. An important area for further work is to explore a setting where resource constraints are present at both training and deployment time. For example, a consideration of the impact of pruning on pre-trained models, such as large multilingual MT models that are known to boost low-resource NMT quality (Aharoni et al., 2019; Arivazhagan et al., 2019). Additionally, the minimal differences observed in our human evaluation of preferences open up a range of questions for deeper qualitative analysis of the resulting translations: Under which conditions do humans notice differences, and how do translations differ in style? There may be interesting connections to recent findings about output hallucinations occurring on memorized ex-

amples (Raunak et al., 2021), or with respect to translation bias (Koppel and Ordan, 2011).

6 Conclusion

We demonstrate the effectiveness of introducing sparsity when training NMT models for low-resourced languages. We show that small performance drops in extremely sparse regimes according to automatic metrics are not reflected in human-judged translation quality. Our extensive study of the impact of pruning on out-of-distribution generalization reveals that sparse models improve over dense models in a limited data regime. Overall, these insights are promising for overcoming the low-resource double bind: Pruned models reduce resource requirements for deployment, and increase the robustness towards out-of-domain samples due to reduced memorization during training.

7 Acknowledgements

References

Kale ab Tessera, Sara Hooker, and Benjamin Rosman. 2021. Keep the gradients flowing: Using gradient flow to study sparse network optimization.

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Alham Fikri Aji and Kenneth Heafield. 2020. Compressing neural machine translation models with 4-bit precision. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 35–42, Online. Association for Computational Linguistics.

Dario Amodei, Danny Hernandez, Girish Sastry, Jack Clark, Greg Brockman, and Ilya Sutskever. 2018. Ai and compute.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Francisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, William Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. TICO-19: the translation initiative for covid-19. *CoRR*, abs/2007.01788.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, X. Jiang, Qun Liu, Michael R. Lyu, and Irwin King. 2020. Binarybert: Pushing the limit of bert quantization. *ArXiv*, abs/2012.15701.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Elan Van Biljon, Arnu Pretorius, and Julia Kreutzer. 2020. On optimal transformer depth for low-resource language translation. *AfricaNLP Workshop*.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*.

Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. 2020. When is memorization of irrelevant training data necessary for high-accuracy learning?

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.

Xiao-Han Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang, and Jing jing Liu. 2021. Earlybert: Efficient bert training via early-bird lottery tickets. *ArXiv*, abs/2101.00063.

Insoo Chung, Byeongwook Kim, Yoonjung Choi, Se Jung Kwon, Yongkweon Jeon, Baeseong Park, Sangha Kim, and Dongsoo Lee. 2020. Extremely low bit transformer quantization for on-device neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4812–4826, Online. Association for Computational Linguistics.

Christopher Cieri, Mike Maxwell, Stephanie Strassel, and Jennifer Tracey. 2016. Selection criteria for low resource language programs. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4543–4549, Portorož, Slovenia. European Language Resources Association (ELRA).

Maxwell D. Collins and Pushmeet Kohli. 2014. Memory Bounded Deep Convolutional Networks. *CoRR*, abs/1412.1442.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2014. Training deep neural networks with low precision multiplications. *arXiv e-prints*, page arXiv:1412.7024.

Yann Le Cun, John S. Denker, and Sara A. Solla. 1990. Optimal brain damage. In *Advances in Neural Information Processing Systems*, pages 598–605. Morgan Kaufmann.

Bonaventure F. P. Dossou and Chris C. Emezue. 2021. Okwugbé: End-to-end speech recognition for fon and igbo. *AfricaNLP Workshop*.

Kevin Duh, Paul McNamee, Matt Post, and Brian Thompson. 2020. Benchmarking neural and statistical machine translation on low-resource African languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2667–2675, Marseille, France. European Language Resources Association.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online. Association for Computational Linguistics.

Erich Elsen, Marat Dukhan, Trevor Gale, and Karen Simonyan. 2019. Fast sparse convnets.

Andre Esteva, Brett Kuprel, Roberto Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. 2019. Rigging the lottery: Making all tickets winners.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems*, volume 33, pages 2881–2891. Curran Associates, Inc.

Trevor Gale, Erich Elsen, and Sara Hooker. 2019. The state of sparsity in deep neural networks. *CoRR*, abs/1902.09574.

Trevor Gale, Erich Elsen, and Sara Hooker. 2019. The State of Sparsity in Deep Neural Networks. *arXiv e-prints*, page arXiv:1902.09574.

Trevor Gale, Matei Zaharia, Cliff Young, and Erich Elsen. 2020. Sparse gpu kernels for deep learning.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Yiwen Guo, Anbang Yao, and Yurong Chen. 2016. Dynamic Network Surgery for Efficient DNNs. In *NeurIPS*.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. Deep learning with limited numerical precision. *CoRR*, abs/1502.02551.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both Weights and Connections for Efficient Neural Network. In *NeurIPS*, pages 1135–1143.

B. Hassibi, D. G. Stork, and G. J. Wolff. 1993a. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pages 293–299 vol.1.

Babak Hassibi, David G. Stork, and Stork Crc. Ricoh. Com. 1993b. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems 5*, pages 164–171. Morgan Kaufmann.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv e-prints*, page arXiv:1503.02531.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. 2020a. What do compressed deep neural networks forget?

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. 2020b. Characterising bias in compressed models.

M. Horowitz. 2014. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 10–14.

Lu Hou, Lifeng Shang, X. Jiang, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. *ArXiv*, abs/2004.04037.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv e-prints*.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Quantized neural networks: Training neural networks with low precision weights and activations. *CoRR*, abs/1609.07061.

F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and 0.5MB model size. *ArXiv e-prints*.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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.

Moshe Koppel and Noam Ordan. 2011. Translationese and its dialects. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1318–1326, Portland, Oregon, USA. Association for Computational Linguistics.

Ashish Kumar, Saurabh Goyal, and Manik Varma. 2017. Resource-efficient machine learning in 2 KB RAM for the internet of things. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1935–1944, International Convention Centre, Sydney, Australia. PMLR.

N. D. Lane and P. Warden. 2018. The deep (learning) transformation of mobile and embedded computing. *Computer*, 51(5):12–16.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Sebastian Ruder, Dani Yogatama, Kris Cao, Tomas Kocisky, Susannah Young, and Phil Blunsom. 2021. Pitfalls of static language modelling.

Bei Li, Ziyang Wang, H. Liu, Quan Du, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2020a. Learning light-weight translation models from deep transformer. *ArXiv*, abs/2012.13866.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joseph E. Gonzalez. 2020b. Train large, then compress: Rethinking model size for efficient training and inference of transformers. *ICML*.

Lucas Liebenwein, Cenk Baykal, Brandon Carter, David Gifford, and Daniela Rus. 2021. Lost in pruning: The effects of pruning neural networks beyond test accuracy. *CoRR*, abs/2103.03014.

C. Louizos, M. Welling, and D. P. Kingma. 2017. Learning Sparse Neural Networks through L_0 Regularization. *ArXiv e-prints*.

Laura Martinus and Jade Z. Abbott. 2019. A focus on neural machine translation for african languages. *CoRR*, abs/1906.05685.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H. Nguyen, Madeleine Gibescu, and Antonio Liotta. 2018. Scalable Training of Artificial Neural Networks with Adaptive Sparse Connectivity Inspired by Network Science. *Nature Communications*.

Kenton Murray, Jeffery Kinnison, Toan Q. Nguyen, Walter Scheirer, and David Chiang. 2019. Autosizing the transformer network: Improving speed, efficiency, and performance for low-resource machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 231–240, Hong Kong. Association for Computational Linguistics.

Sharan Narang, Erich Elsen, Gregory Diamos, and Shubho Sengupta. 2017. Exploring Sparsity in Recurrent Neural Networks. *arXiv e-prints*, page arXiv:1704.05119.

Sharan Narang, Erich Elsen, Gregory Diamos, and Shubho Sengupta. 2017. Exploring sparsity in recurrent neural networks.

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Selinga, Lawrence Okegbemi, Laura Martinus, Kolawole Tajudeen, Kevin Degila, Kelechi Ogueji, Kathleen Siminyu, Julia Kreutzer, Jason Webster, Jamiil Toure Ali, Jade Abbott, Iroro Orife, Ignatius Ezeani, Idris Abdulkadir Dangana, Herman Kamper, Hady Elsahar, Goodness Duru, Ghollah Kioko, Murhabazi Espoir, Elan van Biljon, Daniel Whitenack, Christopher Onyefuluchi, Chris Chinenye Emezue, Bonaventure F. P. Dossou, Blessing Sibanda, Blessing Bassey, Ayodele Olabiyi, Arshath Ramkilowan, Alp Öktem, Adewale Akinfaderin, and Abdallah Bashir. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.

Alp Öktem, Eric DeLuca, Rodrigue Bashizi, Eric Paquin, and Grace Tang. 2021. Congolese swahili machine translation for humanitarian response. *AfricaNLP Workshop*.

Alp Öktem, Mirko Plitt, and Grace Tang. 2020. Tigrinya neural machine translation with transfer learning for humanitarian response. *AfricaNLP Workshop*.

Iroro Orife. 2018. Attentive sequence-to-sequence learning for diacritic restoration of yorùbá language text. *CoRR*, abs/1804.00832.

Edward J. Oughton. 2021. Policy options for digital infrastructure strategies: A simulation model

for broadband universal service in africa. *CoRR*, abs/2102.03561.

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 on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training.

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.

Jerry Quinn and Miguel Ballesteros. 2018. Pieces of eight: 8-bit neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 114–120, New Orleans - Louisiana. Association for Computational Linguistics.

Vikas Raunak, Siddharth Dalmia, Vivek Gupta, and Florian Metze. 2020. On long-tailed phenomena in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3088–3095, Online. Association for Computational Linguistics.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. *CoRR*, abs/2104.06683.

R. Reed. 1993. Pruning algorithms-a survey. *IEEE Transactions on Neural Networks*, 4(5):740–747.

Ravi K Samala, Heang-Ping Chan, Lubomir M Hadji-iski, Mark A Helvie, Caleb Richter, and Kenny Cha. 2018. Evolutionary pruning of transfer learned deep convolutional neural network for breast cancer diagnosis in digital breast tomosynthesis. *Physics in Medicine & Biology*, 63(9):095005.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning.

Abigail See, Minh-Thang Luong, and Christopher D. Manning. 2016. Compression of Neural Machine Translation Models via Pruning. *arXiv e-prints*, page arXiv:1606.09274.

Abigail See, Minh-Thang Luong, and Christopher D. Manning. 2016. Compression of neural machine translation models via pruning. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 291–301, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832, Online. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp.

Nikko Ström. 1997. Sparse connection and pruning in large dynamic artificial neural networks.

Fei Sun, Minghai Qin, Tianyun Zhang, Liu Liu, Yen-Kuang Chen, and Yuan Xie. 2020. Computation on sparse neural networks and its implications for future hardware: Invited. In *Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference*, DAC '20. IEEE Press.

Allahsera Auguste Tapo, Bakary Coulibaly, Sébastien Diarra, Christopher Homan, Julia Kreutzer, Sarah Luger, Arthur Nagashima, Marcos Zampieri, and Michael Leventhal. 2020. Neural machine translation for extremely low-resource African languages: A case study on Bambara. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 23–32, Suzhou, China. Association for Computational Linguistics.

Allahsera Auguste Tapo, Michael Leventhal, Sarah Luger, Christopher M. Homan, and Marcos Zampieri. 2021. Domain-specific MT for low-resource languages: The case of bambara-french. *AfricaNLP Workshop*.

Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. 2020. The Computational Limits of Deep Learning. *arXiv e-prints*, page arXiv:2007.05558.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for Neural Machine Translation. *CoRR*, abs/1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

P. Warden and D. Situnayake. 2019. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media, Incorporated.

W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. 2016. Learning Structured Sparsity in Deep Neural Networks. *ArXiv e-prints*.

Chen Xu, Bojie Hu, Yufan Jiang, Kai Feng, Zeyang Wang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2020. Dynamic curriculum learning for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3977–3989, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xiaowei Zhang, Wei Chen, Feng Wang, Shuang Xu, and Bo Xu. 2017. Towards compact and fast neural machine translation using a combined method. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1475–1481, Copenhagen, Denmark. Association for Computational Linguistics.

Maohua Zhu, Tao Zhang, Zhenyu Gu, and Yuan Xie. 2019. Sparse tensor core: Algorithm and hardware co-design for vector-wise sparse neural networks on modern gpus. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '52, page 359–371, New York, NY, USA. Association for Computing Machinery.

Michael Zhu and Suyog Gupta. 2017. To prune, or not to prune: exploring the efficacy of pruning for model compression. *CoRR*, abs/1710.01878.

G.K. Zipf. 1999. *The Psycho-Biology of Language: An Introduction to Dynamic Philology*. Cognitive psychology]. Routledge.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Training

A.1 Overview of Compression

Popular model compression research directions includes reducing the precision or bit size per model weight (quantization) (Jacob et al., 2018; Courbariaux et al., 2014; Hubara et al., 2016; Gupta et al., 2015), efforts to start with a network that is more compact with fewer parameters, layers or computations (architecture design) (Howard et al., 2017; Iandola et al., 2016; Kumar et al., 2017), student networks with fewer parameters that learn from a larger teacher model (model distillation) (Hinton et al., 2015) and finally pruning by setting a subset of weights or filters to zero (Louizos et al., 2017; Wen et al., 2016; Cun et al., 1990; Hassibi et al., 1993a; Ström, 1997; Hassibi et al., 1993b; See et al., 2016; Narang et al., 2017). Often, a combination of compression methods might be applied. For example, pruning might be combined with other efficiency-improving methods, e.g. quantization or faster search algorithms. Quantization can be used to speed up inference and relax hardware requirements, as has been shown for e.g., 8-bit (Quinn and Ballesteros, 2018), 4-bit (Aji and Heafield, 2020) and recently also below 3-bit quantization (Chung et al., 2020) of NMT models. In the wider NLP space, there has been interest in evaluating the trade-offs of different compression techniques for downstream finetuning. Sanh et al. (2020) propose the use of movement pruning, a simple, deterministic first-order weight pruning method that is more adaptive to pre-trained model fine-tuning.

Magnitude-based weight pruning schemes use the magnitude of each weight as a proxy for its importance to model quality, and remove the least important weights according to some sparsification schedule over the course of training. Many variants have been proposed (Collins and Kohli, 2014; Han et al., 2015; Guo et al., 2016; Zhu and Gupta, 2017), which can be distinguished by differences in the criteria used to remove weights, when weights are removed and whether weights that have been pruned can still receive gradient updates after being removed.

Han et al. (2015) use iterative magnitude pruning and re-training to progressively sparsify a model. The target model is first trained to convergence, after which a portion of weights are removed and the model is re-trained with these weights fixed to zero. This process is repeated until the target

sparsity is achieved. Guo et al. (2016) improve on this approach by allowing masked weights to still receive gradient updates, enabling the network to recover from incorrect pruning decisions during optimization. They achieve higher compression rates and interleave pruning steps with gradient update steps to avoid expensive re-training. Zhu and Gupta (2017) similarly allow gradient updates to masked weights, and make use of a gradual sparsification schedule with sorting-based weight thresholding to maintain accuracy while achieving a user specified level of sparsification.

Magnitude pruning can easily be adapted to induce block or activation level sparsity by removing groups of weights based on their p-norm, average, max, or other statistics. Variants have also been proposed that maintain a constant level of sparsity during optimization to enable accelerated training (Mocanu et al., 2018).

A.2 Architecture Size

Table 5 compares the sizes for base and tiny transformers. We do a model size ablation, comparing two model types (sizes in Appendix Table 5). Table 8 displays the results showing that the `tiny` transformer gives a lower BLEU score than the `base` transformer even with extensive hyperparameter tuning. Hence we use the sparse transformer `base` model for all our preferred experiments.

A.3 Training Hyperparameters

We train the transformer with the hyper-parameters and optimizer settings described in (Vaswani et al., 2017). We use a batch size of 2048, and train on a Google Cloud TPU v2-8 with a default learning rate of 0.2, and learning rate warm-up steps of 8000. Regularization is introduced with dropout and label smoothing with rates 0.1. We are interested in a setting where resource constraints are present at deployment time, and do not assume constraints present at training. Our code is publicly available at https://github.com/orevaahia/mc4lrnmt. In our experiments on the `Full` dataset, models for Yoruba and Igbo and German are trained for a total number of 100k steps while the Hausa models are trained for 60k steps. For our experiments in a data limited regime, we train for a total of 60k steps across all languages.

A.4 Pruning Hyperparameters

We perform a limited hyperparameter search with manual tuning to determine the best pruning start

Hyperparameters	Base	Tiny
Transformer Layers	6	2
Hidden Size	512	128
Attention Heads	8	4
Filter Size	2048	512
Optimizer	adafactor	adafactor
Max Length	64	64

Table 5: Hyperparameters for transformer variants.

time, end time and recovery interval and select the best models based on the BLEU performance, we notice that when we train on the `Full` data, we see an average difference of 0.7 if we introduce pruning early and stop pruning close to the end of training on all sparsity levels we train for. In our case, we begin pruning on the 2000th train step and end pruning on the 80000th step for German, Igbo and Yoruba and begin pruning on the 2000th train step and end pruning on the 40000th for Hausa. This is line with the results reported by Gale et al. (2019) when evaluating different compression techniques on high resourced languages. Training on `limited` data however shows slightly different results. In most cases, we see also see an average increase of 0.7 when we start to prune at nearly a quarter of the total train steps and stop pruning 20,000 steps before training ends. For all experiments on the `Full` data, the frequency of pruning is every 2000 steps, however for `limited` data experiments it is either 1000 or 2000 steps. Full tuning results are in Table 10.

B Human Evaluation Details

Test set quality is evaluated independently, translations for rating were randomly selected from each test set that yielded different translations from the two models. The ratio of identical translations that are withheld from this rating lies around 27% for all languages.

The absolute ratings allow us to draw conclusions about the absolute quality, and the presentation in pairs encourages the rater to consider differences between both translations for their rating. We gather three independent ratings for translations into German, and one rating for translations into Yoruba, Hausa and Igbo. Ratings from multiple raters are aggregated by using the median score.

Table 7 reports absolute scores as well as wins/losses of sentence-level comparisons.

Sparsity	Absolute Ratings		Relative Wins [%]		
	0	90	0	Neither	90
de	4.17	4.06	31	46	23
yo	3.66	3.66	26	48	26
ig	3.87	3.96	31	31	37
ha	4.77	4.85	29	38	34

Table 6: Results of the human evaluation study of 500 `Global` test set translations comparing **dense** (0) and **90%-sparse** (90) models (`Limited`). Absolute ratings are averaged across sentences.

Sparsity	Absolute Ratings		Relative Wins [%]		
	0	90	0	Neither	90
de	3.80	3.66	29	50	21
yo	3.51	3.51	19	61	19
ig	3.81	3.85	35	28	37
ha	4.53	4.53	26	47	27

Table 7: Results of the human evaluation study of 500 `Random` test set translations comparing **dense** (0) and **90%-sparse** (90) models (`Limited`). Absolute ratings are averaged across sentences.

C Distribution Shift Evaluation Results

We provide multiple views on translation quality under distribution shift for all languages in Figures 4 and 5. To ease comparison, we summarize all results for German in Figure 6 since it is present in all the datasets we consider. Relative performance degradation is measured by dividing the BLEU (Papineni et al., 2002) from the sparse models by that of the dense model.

D Results for Full vs Limited Training Regime

Figure 2 shows the relative and absolute differences in BLEU caused by pruning in both `Limited` and `Full` models. We see minimal changes in BLEU between both training regimes and conclude that the ranking of absolute BLEU over all languages doesn't correspond to training data sizes as Yoruba and Igbo; although one-fifth of the `Full` German data still achieves higher BLEU than German.

E Results for In-domain Validation and Test Sets

Figure 7 compares the performance across test and validation sets. We can see that the validation set is closer to the `Random` test set since it was sampled randomly as well. The `Global` test set, however, contains fewer long-tail examples and therefore receives higher BLEU.
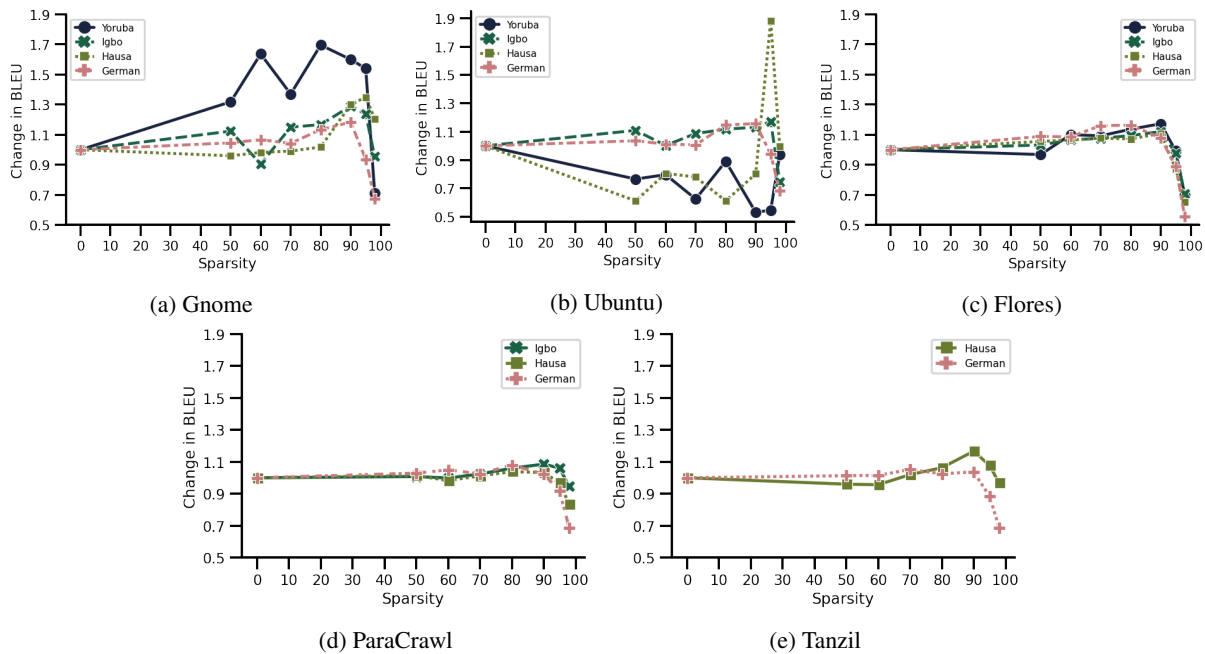
Figure 5: Relative change in BLEU on distribution shift datasets across languages and sparsity levels.
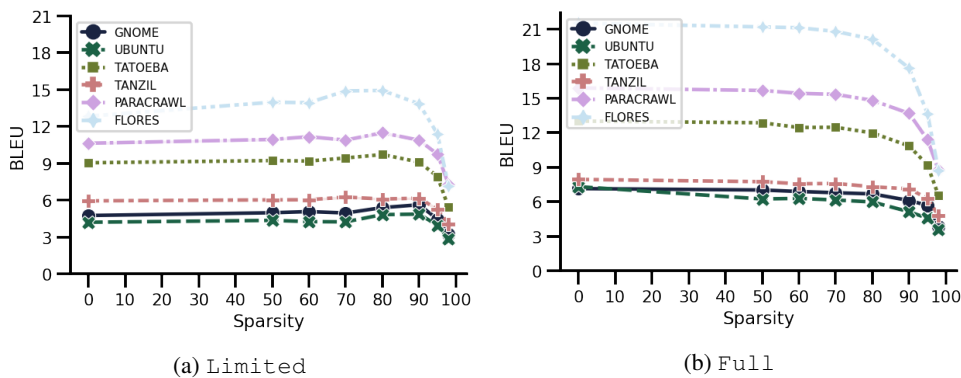

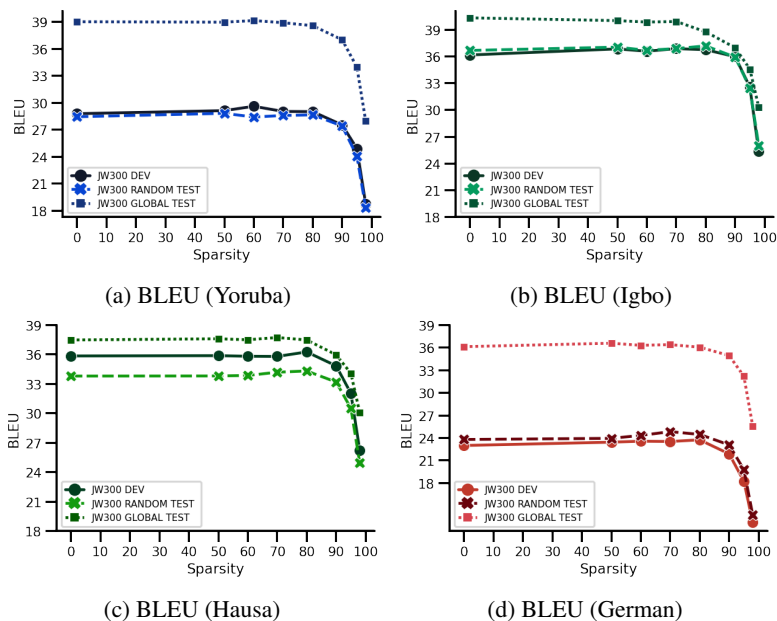Figure 6: Absolute BLEU performance on out-of-distribution data for German across all sparsity levels.

Figure 7: Comparison of Absolute BLEU performance of Random Test, Global Test and Validation sets

Size	%Sparse	# Params	yo		ig		ha		de	
			Random	Global	Random	Global	Random	Global	Random	Global
Tiny	0	14.0M	20.39	30.11	28.79	31.78	27.74	31.84	15.40	28.05
Base	0	46.1M	28.45	39.01	36.68	**40.33**	33.81	37.49	23.80	36.14
	50	23.1M	**28.83**	38.96	37.05	40.02	33.82	37.61	23.95	**36.60**
	60	18.6M	28.42	**39.14**	36.69	39.86	33.87	37.51	24.34	36.32
	70	13.8M	28.59	38.90	36.89	39.92	34.20	**37.74**	**24.84**	36.41
	80	9.4M	28.66	38.57	**37.19**	38.77	**34.33**	37.49	24.47	36.07
	90	4.6M	27.42	37.01	35.93	36.93	33.20	32.05	23.08	34.96
	95	2.5M	24.08	33.97	32.46	34.58	30.52	34.09	19.78	32.21
	98	1.1M	18.34	27.97	26.02	30.32	24.99	30.09	13.78	25.58

Table 8: Number of non-zero parameters and test BLEU scores under the `Limited` training regime.

Size	%Sparse	# Params	yo	ig	ha	de
			Global	Global	Global	Global
Base	0	46.1M	39.01	40.33	37.49	36.14
	50	23.1M	38.96	40.02	37.61	36.6
	60	18.6M	39.14	39.86	37.51	36.32
	70	13.8M	38.90	39.92	37.74	36.41
	80	9.4M	38.57	38.77	37.49	36.07
	90	4.6M	37.01	36.98	35.98	34.96
	95	2.5M	33.97	34.58	34.09	32.21
	98	1.1M	27.97	30.52	30.09	25.58

Table 9: Number of non-zero parameters and test BLEU scores under the `Full` training regime.

%Sparse	BP	EP	PF	yo		ig		ha		de	
				Random	Global	Random	Global	Random	Global	Random	Global
50	2000	60000	1000	28.45	38.82	36.93	39.247	33.81	38.14	24.26	35.77
	12000	40000	2000	28.83	39.07	37.55	39.91	33.95	37.9	24.52	36.6
	15000	40000	2000	28.64	38.77	37.05	40.02	34.02	37.99	23.95	36.25
60	2000	60000	1000	28.28	38.54	37.15	39.61	33.82	38.06	24.16	36.09
	12000	40000	2000	28.42	39.14	36.69	39.86	33.68	37.9	24.3	36.32
	15000	40000	2000	28.54	39.07	37.29	39.58	33.86	37.94	24.34	35.88
70	2000	60000	1000	28.12	38.46	37.55	38.88	34.10	37.84	24.54	36.2
	12000	40000	2000	28.46	38.6	36.89	39.77	33.87	38.29	24.84	36.41
	15000	40000	2000	28.59	38.9	37.32	39.92	33.80	38.15	24.41	36.25
80	2000	60000	1000	28.36	38.07	36.73	38.36	33.72	37.52	23.83	36.02
	12000	40000	2000	28.66	38.57	37.1	38.54	34.41	37.95	24.47	36.07
	15000	40000	2000	28.53	38.19	37.19	38.77	34.33	38.03	24.45	35.82
90	2000	60000	1000	26.50	36.09	35.42	36.73	32.35	36.38	22.68	34.29
	12000	40000	2000	27.42	37.01	35.93	36.98	33.20	36.51	23.08	34.96
	15000	40000	2000	26.96	36.72	36.08	36.91	33.15	36.41	23.23	34.36
95	2000	60000	1000	24.08	32.82	31.49	34.06	29.16	33.32	18.97	31.5
	12000	40000	2000	23.93	33.71	32.46	34.58	30.69	34.39	20.01	32.14
	15000	40000	2000	24.12	33.06	32.44	34.29	30.52	34.61	19.78	32.21
98	2000	60000	1000	18.34	25.95	25.05	28.87	24.22	28.63	12.9	24.35
	12000	40000	2000	17.87	27.47	26.02	30.32	24.99	30.09	14.25	25.58
	15000	40000	2000	17.54	27.28	26.00	29.39	24.99	29.5	13.78	25.13

Table 10: Absolute BLEU performance from tuning pruning hyperparameters across all languages under the `Limited` training regime. **BP=** Begin Pruning Step, **EP=** End Pruning Step and **Pruning Frequency=** Pruning Frequency