Multilingual Brain Surgeon: Large Language Models Can be Compressed Leaving No Language Behind

Hongchuan Zeng[1], **Hongshen Xu**[1], **Lu Chen**[1,2†], **Kai Yu**[1,2†]
[1]X-LANCE Lab, Department of Computer Science and Engineering
MoE Key Lab of Artificial Intelligence, SJTU AI Institute
Shanghai Jiao Tong University, Shanghai, China
[2]Suzhou Laboratory, Suzhou, China
{charlie68, xuhongshen, chenlusz, kai.yu}@sjtu.edu.cn

Abstract

Large Language Models (LLMs) have ushered in a new era in Natural Language Processing, but their massive size demands effective compression techniques for practicality. Although numerous model compression techniques have been investigated, they typically rely on a calibration set that overlooks the multilingual context and results in significant accuracy degradation for low-resource languages. This paper introduces Multilingual Brain Surgeon (MBS), a novel calibration data sampling method for multilingual LLMs compression. MBS overcomes the English-centric limitations of existing methods by sampling calibration data from various languages proportionally to the language distribution of the model training datasets. Our experiments, conducted on the BLOOM multilingual LLM, demonstrate that MBS improves the performance of existing English-centric compression methods, especially for low-resource languages. We also uncover the dynamics of language interaction during compression, revealing that the larger the proportion of a language in the training set and the more similar the language is to the calibration language, the better performance the language retains after compression. In conclusion, MBS presents an innovative approach to compressing multilingual LLMs, addressing the performance disparities and improving the language inclusivity of existing compression techniques. The codes are available at: https://github.com/X-LANCE/MBS.

Keywords: Large Language Model, Multilingual Model Compression

1. Introduction

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) with their remarkable performance. However, their colossal size and computational demands necessitate effective Model Compression (MC) techniques for practical use. In the case of multilingual LLMs, the vast size is crucial for retaining information from various languages and mitigating the curse of multilinguality (Conneau et al., 2020; Goyal et al., 2021). Moreover, wide language coverage and interference among languages pose a harder challenge for compressing multilingual LLMs.

Existing approaches for MC have predominantly focused on model quantization (Frantar et al., 2023; Dettmers et al., 2022; Xiao et al., 2023; Yao et al., 2022), where model parameters are mapped to lower bit-level representations, and network pruning, which reduces the size of neural networks by eliminating unnecessary connections. Inspired by the classic Optimal Brain Damage (OBD) and Optimal Brain Surgeon (OBS) pruning framework (Hassibi et al., 1993; Le Cun et al., 1989), various approaches, namely GPTQ (Frantar et al., 2023) for model quantization, SparseGPT (Frantar and Alistarh, 2023) and Wanda (Sun et al., 2023) for network pruning, have been proposed to compress LLMs. These compression methods utilize a calibration dataset to determine the priority of parameters and thus are retraining-free, avoiding expensive fine-tuning cost especially for LLMs.

However, neither of these methods has considered the multilingual scenario: all of them use a single-language (e.g., English) calibration dataset to determine the priority of parameters for multilingual models. A significant performance drop on multilingual tasks is observed due to this English-centric approach, especially in the case of low-resource languages.

In this paper, we propose Multilingual Brain Surgeon (MBS), which has successfully achieved significant sparsity levels when compressing multilingual LLMs while simultaneously minimizing the performance drop across different languages in the models, leaving no language behind after compression. Specifically, as shown in Figure 1, MBS samples the calibration data of different languages proportionally to the language distribution of the model training dataset. This approach effectively addresses the multilingual compression problem compared to previous monolingual sampling methods. Furthermore, we observed the dynamics of language interaction during compression and drew two main conclusions: 1) *The larger the proportion of a language in the model training dataset*, the more resistant it is to compression. 2) *The*

†Lu Chen and Kai Yu are the corresponding authors.

LREC-COLING 2024, pages 11794–11812
20-25 May, 2024.

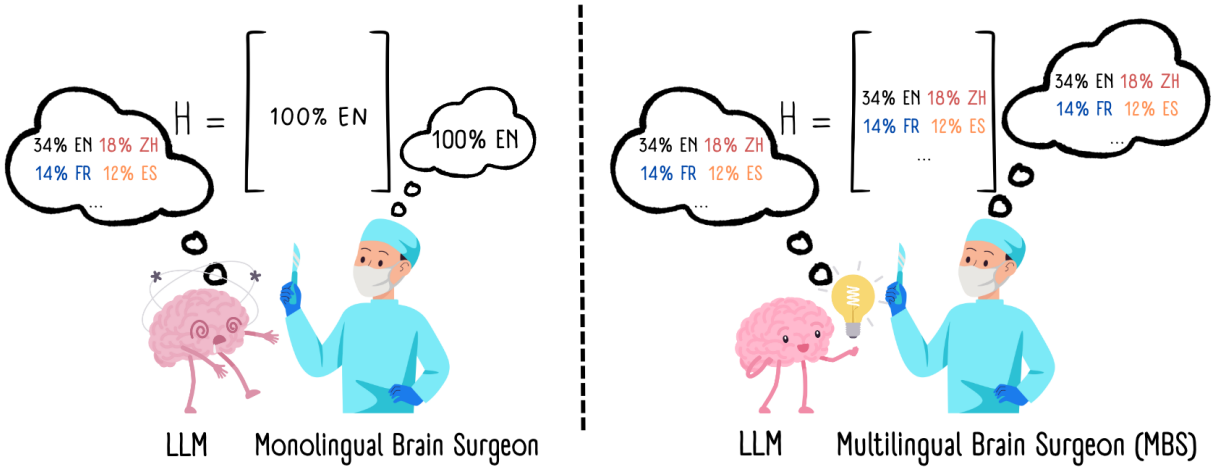

Figure 1: MBS samples calibration data from different languages proportionally to the language distribution of training datasets. This approach (right part) effectively addresses the multilingual compression problem compared to previous monolingual sampling methods (left part).

more similar the downstream language is to the calibration language, the less performance drop it obtained after compression. We further propose a measure of similarity among languages to explain and predict the performance drop.

The experiments were conducted on BLOOM (BigScience Workshop, 2022), one of the most effective open-source multilingual LLM models. We sample the calibration data from CC-100 (Wenzek et al., 2020), a widely used dataset of web-crawled data containing 100+ languages. The perplexity of languages is tested on XL-Sum (Hasan et al., 2021), a dataset that contains high-quality articles from BBC covering 45 languages. Experimental results demonstrate that MBS enhances the performance of GPTQ, SparseGPT, and Wanda compared to using only English calibration data. We want to further highlight that MBS is applicable to all compression methods that involve the use of calibration data, especially those following the OBS/OBD framework (Hassibi et al., 1993; Le Cun et al., 1989), which necessitates approximations of second-derivative information.

2. Background

2.1. Optimal Brain Surgeon (OBS)

Optimal Brain Surgeon (Hassibi et al., 1993) is a classic network pruning algorithm. It assumes that a network's error converges to a local minimum and calculates the second-order derivatives (Hessian matrix $\mathbf{H}$) of the error (E) with respect to each parameter (w) to determine which connections can be safely pruned without significantly affecting performance. The increase in error (L_j) when a parameter (w_j) is set to zero, and the optimal adjustment (δw) of the remaining weights to compensate for the removal are given by:

$$L_j = \frac{1}{2}\frac{{w_j}^2}{[\mathbf{H}^{-1}]_{jj}} \tag{1}$$

$$\delta w = -\frac{w_j}{[\mathbf{H}^{-1}]_{jj}}\mathbf{H}^{-1}_{:,j}. \tag{2}$$

2.2. Error Measurement

The network's error can be expressed in terms of the l_2-error between the outputs before and after compression (Hubara et al., 2021). Given inputs $\mathbf{X}$ (the training dataset), the original weights $\mathbf{W}$, the updated weights $\hat{\mathbf{W}}$, and a sparsity mask $\mathbf{M}$ of the same size as $\mathbf{W}$, the error is defined as:

$$E = ||\mathbf{W}\mathbf{X} - (\mathbf{M} \odot \hat{\mathbf{W}})\mathbf{X}||_2^2. \tag{3}$$

In the case of quantization, the mask is a matrix filled with ones. The second-order derivatives ($\mathbf{H}$) of the error with respect to the parameters are therefore represented as $\mathbf{H} = \mathbf{2X}\mathbf{X}^\mathbf{T}$, which forms the basis of our approximation objective.

2.3. SparseGPT, Wanda and GPTQ

To assess the importance of parameters, SparseGPT and Wanda employ different pruning metrics. Taking inspiration from OBS, SparseGPT defines its metric as $S_{i,j} = [|\mathbf{W}|^2/\mathbf{diag}((\mathbf{X}^\mathbf{T}\mathbf{X} + \lambda\mathbf{I})^{-1})]_{i,j}$, with λ being the Hessian dampening factor to prevent inverse computation collapse. On the other hand, Wanda uses $S_{i,j} = |\mathbf{W}_{i,j}| \cdot ||\mathbf{X}_j||_2$ as its pruning metric.

Remarkably, these two metrics are essentially equivalent when λ is set to 0, and only the diagonal elements of the Hessian matrix $X^TX + \lambda I$ are

retained:

$$\mathbf{diag}(((\mathbf{X}^\mathbf{T}\mathbf{X} + \lambda\mathbf{I}) \odot \mathbf{I})^{-1}) = (||\mathbf{X}_j||_2^2)^{-1}. \quad (4)$$

This assumption aligns with the practice of Optimal Brain Damage (Le Cun et al., 1989), which retains only the diagonal elements of the second-order derivatives matrix. Consequently, we can conclude that:

$$\mathbf{S}_{SparseGPT} = \mathbf{S}^2_{Wanda} \quad (5)$$

if we disregard the non-diagonal elements of $\mathbf{H}$.

The primary distinctions between SparseGPT and Wanda are as follows:

- SparseGPT retains the non-diagonal elements of the Hessian metrics, whereas Wanda takes the opposite approach.
- SparseGPT performs adjustments (δw) on non-pruned parameters to compensate for removal, while Wanda does not.

Equally inspired by OBD, the quantization formulas provided by GPTQ are as follows:

$$w_j = \mathbf{argmin}_{w_j} \frac{(\mathbf{quant}\,(w_j) - w_j)^2}{[\mathbf{H}^{-1}]_{jj}} \quad (6)$$

$$\boldsymbol{\delta}_F = -\frac{w_j - \mathbf{quant}\,(w_j)}{[\mathbf{H}^{-1}]_{jj}} \cdot (\mathbf{H}^{-1})_{:,j} \quad (7)$$

Here, w_j represents the greedy-optimal weight to quantize next, $\boldsymbol{\delta}_F$ denotes the corresponding optimal update of weights, and $\mathbf{quant}(w)$ rounds the value of w to the nearest point on the quantization grid. It's evident that these formulas follow a similar pattern to the OBD/OBS approach, and the information of the Hessian matrix $\mathbf{H}$ is crucial in all these methods.

3. Is Monolingual Calibrating Applicable to Multilingual MC?

Previous model compression methods only use English corpus as the sole calibration data, neglecting other languages. This raises the question: **how does monolingual calibration impact the performance of other languages during multilingual model compression?** In this section, we aim to explore this issue theoretically, focusing on two main aspects: the proportion of languages in the training data, and the similarity between languages. Further experimental analysis will be provided in Section 5.3.

We denote the total error of the model as E, and the error on language m as E_m. We know that model training convergence applies to the whole training dataset. Thus, E resides in a local minimum. However, for languages m and n, E_m and E_n may not necessarily be in their own local minima. This also explains the presence of the multilingual curse (Conneau et al., 2020), where the performance of a multilingual model in all languages is lower than that of a monolingual model with the same configuration. This occurs because the model is in a global local minimum, rather than individual local minima for each language. Due to their differing distributions, the local minima for each language do not overlap. The reason why using larger language models can alleviate this problem might be that, with a huge amount of parameters, they can simulate a distribution sophisticated enough where different languages' local minima are close.

3.1. Proportion in training data

Due to the fact that the size of English corpus is much larger than low-resource languages, we may suppose a language pair m and n with a significantly different training corpus size ($p_n >> p_m$).

Intuitively, we can assume that languages with larger corpora in the training set tend to have their minimum error closer to the minimum of E because they contribute more weight to the total error. This characteristic makes them more robust against compression. Conversely, languages with smaller corpora inherently have their minimum error farther from the minimum of E, and compression can potentially push them even further away.

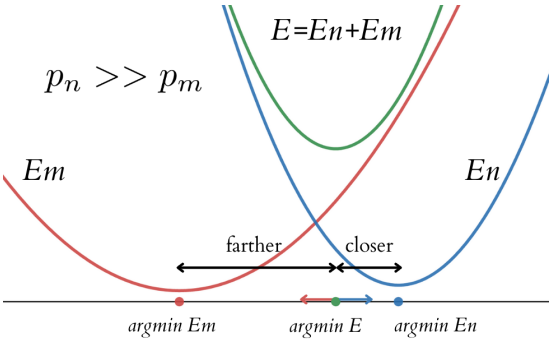

Figure 2: Languages with larger corpora have their minimum error closer to the minimum of E. Monolingual compression effectively "pushed" the model's state towards the minimum error of that particular language.

This phenomenon is manifested in the following way illustrated in Figure 2: *when compressing models with only the calibration data of the well-represented*[1] *language n, it has a significant impact on the performance of the underrepresented language m. However, compressing models with only*

[1] In the rest of the paper, we call a language "well-represented" when its proportion is relatively big in the model training set, and "underrepresented" when its proportion is relatively small.

the calibration data of the underrepresented language m has a comparatively minor impact on the performance of the well-represented language n.

3.2. Similarity between languages

In the second scenario, we may suppose that the two languages are as well-represented as each other ($p_m \approx p_n$). According to Equation 1, the priority of compression is fully determined by $\mathbf{H}$, so it is sufficient to compare $\mathbf{H_m}$ and $\mathbf{H_n}$. We may suppose the non-diagonal elements are trivial (Le Cun et al., 1989) to calculate the inverse of $\mathbf{H}$. The metric is thus simplified to $S = |\mathbf{W}| \cdot ||\mathbf{X}||_\mathbf{2}$, so we can directly compare $||\mathbf{X}||_2$, which is a vector of length q (number of parameters), and each of the elements is the sum of the square of the inputs at the corresponding position.

A classic method to compare the similarity of two vectors is cosine similarity. The choice of cosine similarity over Euclidean distance is motivated by the need to compare two vectors based on the likelihood that their largest components remain consistent after undergoing the same element-wise multiplication with unknown vectors (model parameters). This can be modeled as the comparison of two vectors after they have experienced the same coordinate axis transformation, assessing whether their largest components remain identical. Clearly, when two vectors have a smaller angle between them, the likelihood that their largest components remain the same after undergoing the same coordinate axis transformation is relatively higher (demonstrated in Figure 3).

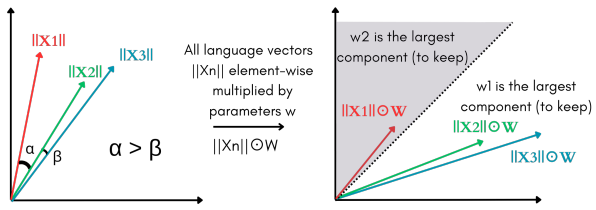

Figure 3: The angle between language 2 and language 3 is smaller than that between language 1 and language 2. After element-wise multiplication, language 2 and 3 are more likely to prioritize the same parameter w_1 because their angle before multiplication is smaller.

However, it's important to acknowledge that cosine similarity does not fulfill the properties of a distance metric, particularly the triangle inequality. Consequently, we cannot directly deduce the similarity between languages 1 and 3 from the similarities between 1 and 2, and 2 and 3. However, the property of a distance metric is less critical in the context of our work, since our goal is only to compare the similarity between the calibration language and the non-calibration languages, rather than among non-calibration languages.

We can compute the cosine similarity between $||\mathbf{X}_m||_2$ and $||\mathbf{X}_n||_2$. When they are similar, using only data of language m as calibration data will introduce little performance drop in language n, and vice versa. *That is to say, when two languages are very different, employing data from just one of the two languages as calibration data will lead to a significant performance decrease in the other.*

4. Multilingual Brain Surgeon (MBS)

To mitigate interference among languages in multilingual model compression, we introduce Multilingual Brain Surgeon (MBS), a method that proportionally samples calibration data from different languages based on their distribution in the model training dataset. We provide additional theoretical details as follows.

In the OBD/OBS framework, we treat the error (E) as a whole. This makes sense for monolingual models since they contain only one language. However, for multilingual models, the error can be regarded as the sum of errors (E_n) associated with different languages. For a model trained on multiple languages, we can express the total error as follows:

$$E = E_1 + E_2 + E_3 + \ldots + E_n. \tag{8}$$

Consequently, the Hessian matrix can be represented as the sum of Hessian matrices for each language:

$$\mathbf{H} = \mathbf{H_1} + \mathbf{H_2} + \mathbf{H_3} + \ldots + \mathbf{H_n}, \tag{9}$$

where $\mathbf{H_n} = \mathbf{X_n}^\mathbf{T}\mathbf{X_n}$. Here, $\mathbf{X_n}$ represents the inputs (training data) for language n, with a shape of $q \times p_n$, where q is the total number of network parameters, and p_n is the total number of training samples for language n.

Let's denote a subset of training data as $X_n^{[k]}$. Then, we have:

$$\mathbf{H_n} = \mathbf{X_n}^\mathbf{T}\mathbf{X_n} = \sum_{k=1}^{p_n} {X_n^{[k]}}^T X_n^{[k]}, \tag{10}$$

which leads to:

$$\mathbf{H} = \sum_{k=1}^{p_1} {X_1^{[k]}}^T X_1^{[k]} + \sum_{k=1}^{p_2} {X_2^{[k]}}^T X_2^{[k]} + \ldots + \sum_{k=1}^{p_n} {X_n^{[k]}}^T X_n^{[k]}. \tag{11}$$

It's evident that each language's contribution to $\mathbf{H}$ depends on its representation in the model's training data. Therefore, when selecting calibration data, it's essential to choose samples from each language in proportion to its presence in the training

set. Specifically, for language n, the percentage of its representation in the training set is p_n/p, where p is the total number of training samples. *Thus, we should allocate a proportionate amount of data from language n (i.e., p_n/p percent) in the calibration data used for compression.*

5. Experiments

5.1. Experimental Setup

Models. The experiments were conducted using the BLOOM (BigScience Workshop, 2022) model family, which is recognized as one of the most effective open-source multilingual LLMs. Our primary tests were performed on both the BLOOM-560m and BLOOM-7b1 models to provide insights into the performance of smaller and larger models. For the network pruning experiments, a pruning sparsity of 50% was applied. In the quantization experiments, the models were quantized to 3 bits precision with groupings of size 1024.

Datasets & Language Selection. For calibration data, we selected CC-100 (Wenzek et al., 2020), a dataset comprising web-crawled content in over 100 languages, similar to the setup used by previous studies like Frantar and Alistarh (2023), Sun et al. (2023), and Frantar et al. (2023) which used a monolingual English dataset called C4 (Raffel et al., 2019).

To evaluate multilingual perplexity, we employed XL-Sum (Hasan et al., 2021), a dataset containing high-quality articles from BBC covering 45 languages, as our benchmark. Additionally, we assessed perplexity on the test sets of raw-WikiText2 (Merity et al., 2016), a widely used English perplexity benchmark. Due to resource limitations for certain languages in the BLOOM model, we conducted experiments on a subset of 20 languages, which were those available in CC-100, XL-Sum, and BLOOM. These languages include Arabic (`ar`), Bengali (`bn`), Chinese simplified (`zh-Hans`), Chinese traditional (`zh-Hant`), French (`fr`), Gujarati (`gu`), Hindi (`hi`), Igbo (`ig`), Indonesian (`id`), Marathi (`mr`), Nepali (`ne`), Portuguese (`pt`), Spanish (`es`), Swahili (`sw`), Tamil (`ta`), Telugu (`te`), Urdu (`ur`), Vietnamese (`vi`), and Yoruba (`yo`).

Evaluation. We evaluated the perplexity of the compressed model separately for each language using XL-Sum. We also conducted zero-shot evaluations, employing the widely recognized EleutherAI-eval-harness (Gao et al., 2021), with a focus on multilingual tasks to assess the performance of less-represented languages. The zero-shot tasks that we have chosen to evaluate the compressed model are specified in Table 1.

Calibration data & Baselines. Our calibration data consisted of 256 segments, each containing 2048 tokens, sampled from CC-100. We used our MBS sampling method and sampled 87, 47, 37, 31, 14, 13, 7, 4, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 segments respectively for `en`, `zh-Hans`, `fr`, `es`, `pt`, `ar`, `vi`, `hi`, `id`, `bn`, `ta`, `te`, `ur`, `ne`, `mr`, `gu`, `zh-Hant`, `sw`, `yo` and `ig`. Additionally, we conducted tests in the *Equal MBS* setting, in which an equal number of segments were sampled from each language. We also implemented the monolingual compression setting, using 256 segments from the same language.

Language Similarity Study. To study language similarity, we conducted monolingual pruning on English and Igbo, representing the best-represented and worst-represented languages in our dataset, respectively. We also performed similar experiments on Urdu and Tamil, which respectively represent the least and most similar languages to the others (further explanation is provided in the results section). To compare language similarity, we utilized the representations after the embedding layer of the BLOOM model, as the compression algorithms do not affect the embedding layer.

5.2. Main results

We conducted our MBS sampling technique to compress both BLOOM-7b1 and BLOOM-560m models, using GPTQ, SparseGPT, and Wanda. The trends observed in the results for these two models are similar. For the sake of better formatting, we will present the results for the 7b1 model in the main text and provide the results for the 560M model in the appendices.

Perplexity. Figure 4 presents the evaluation of perplexity for each language after compression on the BLOOM-7b1 model. The baselines consist of monolingual compression using English-only calibration data.

1. Across **various compression methods**, the MBS sampling technique consistently leads to minimal increases in perplexity. This holds true whether we utilize Wanda or SparseGPT for pruning or GPTQ for quantization.
2. For **underrepresented languages** (located on the right side of the axis), MBS can notably reduce the increase in perplexity after compression, thus preserving the model's capacity for lower-resourced languages, **leaving no languages behind**.
3. Even for **the most well-represented language**, specifically English (on both datasets "en" and "wikitext2"), using MBS sampling introduces a **lower** perplexity than its monolingual English-centric sampling counterpart.

Zero shot tasks. Table 1 provides an overview of the performance of zero-shot tasks after the

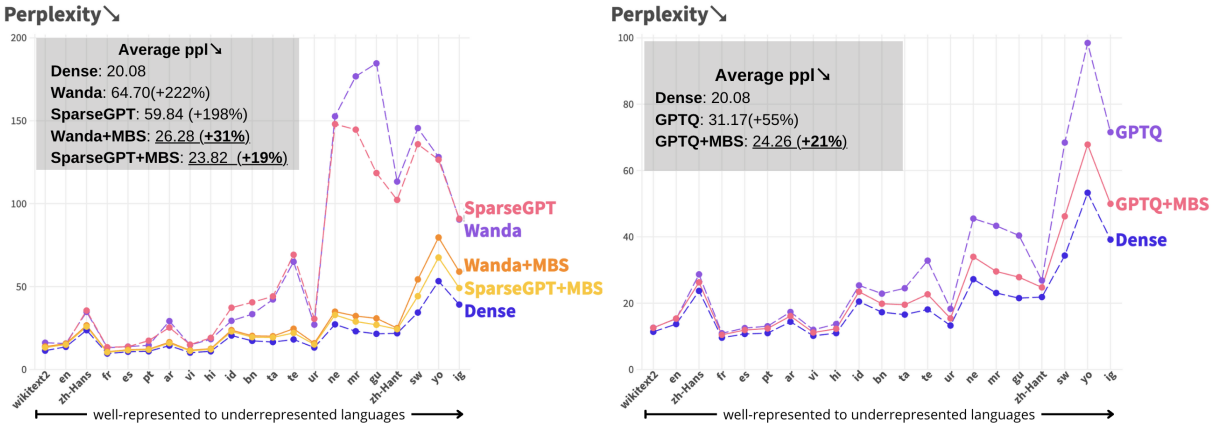

Figure 4: Perplexity for each language and their respective increases when compared to the dense BLOOM-7b1 model after pruning (left) or quantization (right). From left to right, languages are ranked in order from the most well-represented to the least represented.

compression process. The results demonstrate that, in the majority of tasks, utilizing MBS sampling yields superior performance compared to other sampling techniques. Furthermore, the performance after compression closely approximates that of the dense model, highlighting the effectiveness of our approach.

Equal MBS. Table 2 provides the results for *Equal MBS*, where an equal number of samples are taken from each language. While Equal MBS is not the optimal setting, it generally improves the performance of the compressed model. This demonstrates that even without access to the distribution of languages in the training set, Equal MBS can still enhance compression results for the chosen languages, showcasing the versatility of our method.

5.3. Monolingual Compression Study

5.3.1. Factor 1: Proportion in training data

To investigate how the proportion of a language in the training data affects compression results, we selected English (`en`) and Igbo (`ig`), which have the largest and smallest proportions in the training data among the languages in our experiments, respectively. The results are presented in Figure 5.

It is evident that if we use only English as our calibration data, it significantly impacts less well-represented languages, causing substantial increases in perplexity, particularly for Marathi (`mr`) and Gujarati (`gu`). However, for better-represented languages, English (`en`) has a relatively smaller influence, as observed with Chinese simplified (`zh-Hans`), French (`fr`), and Spanish (`es`). Conversely, when we use only Igbo as our calibration data, the increase in perplexity for the other languages is relatively small. **Clearly, languages with a lower representation in the training set tend to experience a more substantial increase in perplexity.**

5.3.2. Factor 2: Similarity between languages

We calculated the cosine similarity of $||\mathbf{X}_n||_2^2$ for different languages using BLOOM-7b1, and then converted this similarity into degrees. This allowed us to create a distance map between languages. To visualize the relative positions of different languages, we employed Multidimensional scaling (Mead, 1992) and generated a 2-dimensional figure (Figure 6). The original distance map is included in the appendices.

Upon observing this graph, we can identify some interesting clusters. Typically, languages from different language families tend to form distinct clusters. For instance, there is a cluster comprising Indo-European languages such as English, Spanish, French, and Portuguese, a cluster for Chinese simplified and Chinese traditional, both of which are Chinese languages, and another cluster consisting of Niger-Congo languages like Yoruba, Igbo, and Swahili. This clustering may be attributed to the following factors:

- **Shared Grammar Structure**: Languages within the same language family often share similar grammar structures.
- **Shared Tokens**: During the tokenization process, these languages frequently share tokens, including prefixes, suffixes, and other word-building elements.

To investigate how language similarity impacts compression outcomes, we chose to examine two extreme cases: Tamil (`ta`), which is the language that is "closest" to all other languages (with an average distance of 7.25), and Urdu (`ur`), which is the language that is "farthest" from all other languages (with an average distance of 15.45). The results for Wanda are displayed in Figure 7, while the results for SparseGPT, which exhibit similar patterns to those of Wanda, are provided in the appendices.

Accuracy of 0-shot task	Dense	Wanda	Wanda +MBS	SparseGPT	SparseGPT +MBS	GPTQ	GPTQ +MBS
xcopa↑							
id	69.80%	67.20%	***67.40%***	65.60%	***66.40%***	67.20%	***67.40%***
sw	51.60%	54.80%	53.80%	55.20%	51.20%	54.60%	***55.00%***
ta	59.20%	61.20%	57.80%	60.60%	58.60%	58.40%	57.80%
vi	70.80%	69.80%	67.20%	66.80%	66.40%	67.00%	***68.20%***
zh	65.20%	62.00%	***63.60%***	62.20%	***63.80%***	61.00%	***62.60%***
Average	63.32%	63.00%	61.96%	62.08%	61.28%	61.64%	***62.20%***
xstory_cloze↑							
ar	58.57%	53.94%	***54.93%***	54.93%	***56.32%***	56.45%	***57.18%***
en	70.75%	68.23%	67.70%	69.23%	68.96%	68.70%	***68.96%***
es	66.12%	64.39%	63.20%	62.87%	***64.39%***	64.53%	***64.79%***
hi	60.56%	56.92%	***57.18%***	57.64%	***58.44%***	58.04%	***58.17%***
id	64.46%	59.96%	***60.29%***	59.23%	***61.81%***	60.89%	***62.54%***
sw	53.94%	50.89%	***51.69%***	50.69%	***52.02%***	52.28%	***52.95%***
te	57.45%	56.52%	***56.72%***	56.78%	***57.97%***	57.18%	***57.71%***
zh	61.88%	58.37%	***59.56%***	57.91%	***60.89%***	60.03%	***60.03%***
Average	61.71%	58.65%	***58.91%***	58.66%	***60.10%***	59.76%	***60.29%***
xwinograd↑							
en	82.15%	79.40%	78.88%	80.09%	79.74%	79.35%	***79.57%***
fr	71.08%	71.08%	67.47%	72.29%	***73.49%***	65.06%	***67.47%***
pt	76.81%	74.14%	***75.29%***	71.48%	***74.14%***	69.20%	***72.24%***
zh	74.40%	74.40%	***75.79%***	74.40%	***75.20%***	71.23%	***73.81%***
Average	76.11%	74.76%	74.36%	74.57%	***75.64%***	71.21%	***73.27%***
pawsx↑							
en	61.30%	53.60%	***54.75%***	57.50%	***58.25%***	56.75%	***58.60%***
es	59.35%	51.75%	***54.05%***	54.10%	***56.60%***	57.95%	56.10%
fr	50.90%	47.45%	46.45%	50.85%	47.10%	52.30%	48.60%
zh	47.35%	45.05%	***45.45%***	45.70%	***47.45%***	49.10%	***50.00%***
Average	54.73%	49.46%	***50.18%***	52.04%	***52.35%***	54.03%	53.33%
xnli↑							
ar	33.83%	33.67%	***33.91%***	34.89%	34.51%	33.67%	***34.75%***
en	53.91%	52.20%	***52.59%***	53.49%	***53.49%***	52.73%	***52.93%***
es	48.70%	48.14%	47.47%	45.13%	***46.81%***	46.63%	***47.54%***
fr	49.68%	43.57%	***48.38%***	46.29%	***49.00%***	48.58%	***48.62%***
hi	46.51%	42.63%	***44.51%***	40.60%	***45.97%***	44.19%	***46.63%***
sw	37.92%	38.36%	37.80%	37.35%	36.29%	36.63%	***37.33%***
ur	42.10%	39.82%	***40.54%***	40.42%	39.58%	38.42%	***41.98%***
vi	47.05%	45.99%	***46.35%***	42.46%	***44.89%***	44.29%	***46.09%***
zh	35.43%	35.31%	33.99%	34.57%	34.21%	35.27%	34.71%
Average	43.90%	42.19%	***42.84%***	41.69%	***42.75%***	42.27%	41.35%
Average↑	57.63%	55.36%	***55.49%***	55.38%	***56.13%***	55.59%	***57.08%***

Table 1: 0-shot task performance of BLOOM-7b1 with different model compression methods.

It is evident that when we use Tamil (`ta`) as our sole calibration data, the increase in perplexity for other languages is relatively small, especially for languages that are "closer" to Tamil, such as Bengali (`bn`) and Hindi (`hi`). Conversely, when Urdu (`ur`) serves as our sole calibration data, the increase in perplexity for other languages is relatively significant on average. The consistent pattern across all four graphs reveals that **languages more distant from the language being compressed tend to exhibit a more significant increase in perplexity.**

An intriguing case study can be conducted on Chinese simplified and Chinese traditional. Despite their close proximity on the language map, they significantly differ in corpus size. Chinese simplified enjoys a much larger proportion, resulting in a more pronounced impact on Chinese traditional after the compression process, while Chinese simplified remains relatively unaffected. These experiments demonstrate the validity and accuracy of our theory.

6. Related Work

Large Language Model. Large Language Models(Zhao et al., 2023) like GPT-4(OpenAI et al., 2024), LLaMA(Touvron et al., 2023) and OPT(Zhang et al., 2022), which have revolutionized Natural Language Processing through their ability

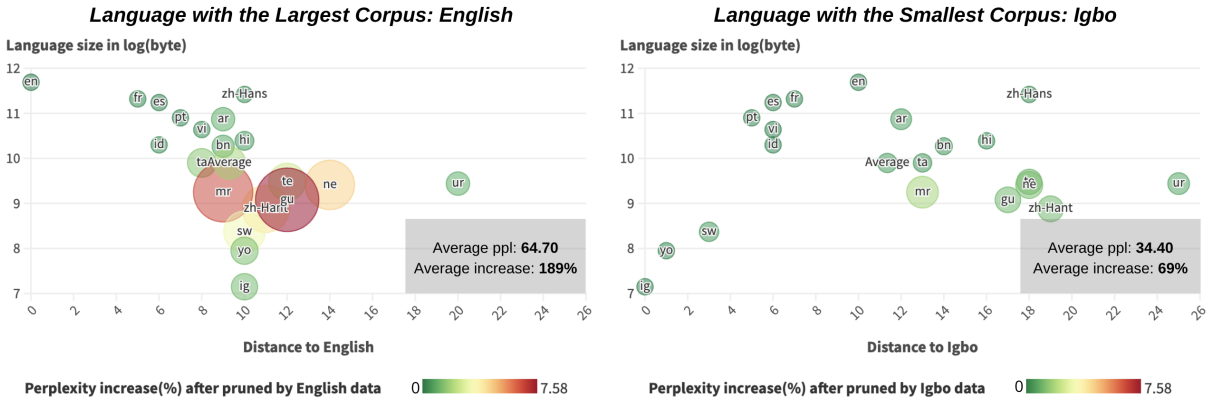

Figure 5: Monolingual pruning results using Wanda with calibration data in English or Igbo. The size of each bubble corresponds to the magnitude of the increase in perplexity for the model in that particular language, while the vertical axis represents the size of training data in log(bytes) from the language in the training set of BLOOM. **The languages with a smaller proportion in the training set experience a greater increase in perplexity.**

Compression Methods	Average 0-shot Task Accuracy↑	Average ppl↓
Wanda	55.36%	64.70
Wanda+ Equal MBS	55.20%	*24.97*
Wanda+ MBS	***55.49%***	***26.28***
SparseGPT	55.38%	59.84
SparseGPT+ Equal MBS	*55.86%*	*22.62*
SparseGPT+ MBS	***56.13%***	***23.82***
GPTQ	55.59%	31.17
GPTQ+ Equal MBS	*56.52%*	*23.15*
GPTQ+ MBS	***57.08%***	***24.26***

Table 2: Performance of *Equal MBS*, where an equal number of segments are sampled from each language.

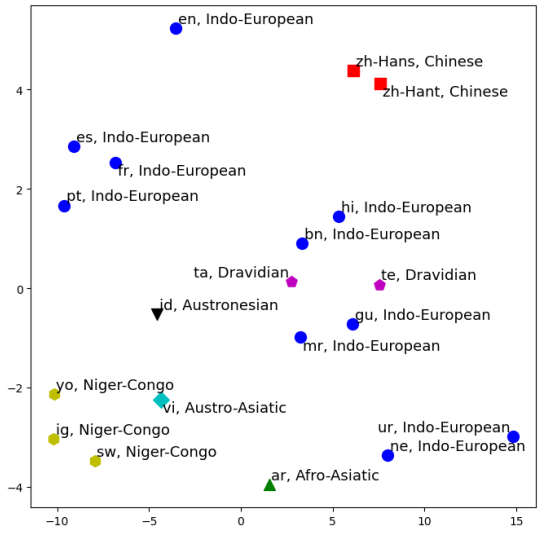

Figure 6: Distance map of different languages associated with their corresponding language families. We can see that languages with the same family cluster together from this map.

to understand and generate nuanced text. Alongside, multilingual language models(Doddapaneni et al., 2021) such as BLOOM(BigScience Workshop, 2022) and XLM-R(Conneau et al., 2020) are breaking language barriers by learning universal representations from texts across numerous languages. These developments underscore a significant shift towards creating more versatile and inclusive NLP systems, with research focusing on architectural innovations, training efficiencies, and cross-lingual capabilities to enhance global digital interaction.

We would like to emphasize that *MBS can be applied to any model compression method that utilizes calibration data, particularly methods based on the OBS/OBD framework*, where the approximation of second-derivative information is required. Thanks to a survey on model compression for large language models by (Zhu et al., 2023), we examined the state-of-the-art model compression methods for large language models, and we found that our MBS is useful for almost all of them.

Pruning and quantization are two major model compression methods for LLMs.

Pruning. Pruning reduces model size and complexity by eliminating unnecessary or redundant components. It can be categorized into **structured pruning**, where higher-granularity structures like rows or columns of weight matrices are removed, and **unstructured pruning**, which eliminates in-

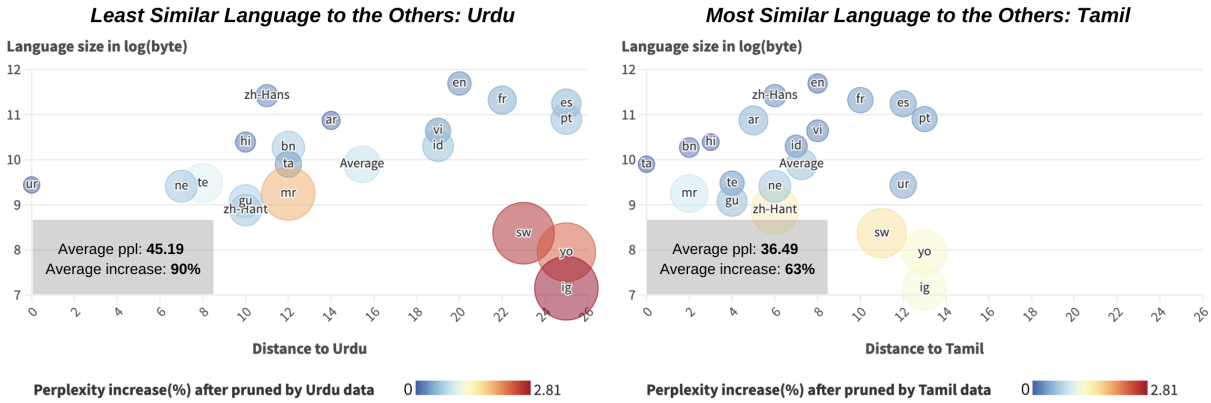

Figure 7: Similarly to Figure 5, but focusing on Urdu or Tamil. **The languages less similar to the calibration language experience a greater increase in perplexity.**

dividual weights, leading to irregular sparse structures. In the domain of **unstructured pruning**, MBS can be applied to Wanda (Sun et al., 2023) and SparseGPT (Frantar and Alistarh, 2023) that we presented in the background section, and also LoRAPrune (Zhang et al., 2023). In the **structured pruning** domain, MBS can empower LLM-Pruner (Ma et al., 2023).

Quantization. Quantization involves converting floating-point numbers into lower bit-level representations, integers, or other discrete forms and can be categorized into **Quantization-Aware Training** and **Post-Training Quantization**. MBS finds numerous applications in quantization, particularly in **post-training quantization**. In post-training quantization, certain approaches focus on quantizing **only the weights** of LLMs. Among these methods, MBS can be applied to AWQ (Lin et al., 2023), GPTQ (Frantar et al., 2023), OWQ (Lee et al., 2023), SpQR (Dettmers et al., 2023), SqueezeLLM (Kim et al., 2023), QuIP (Chee et al., 2023), and SignRound (Cheng et al., 2023). Some other methods try to quantize **both weights and activations** of LLMs. Among them, MBS can be applied to SmoothQuant (Xiao et al., 2023), RPTQ (Yuan et al., 2023), OliVe (Guo et al., 2023), ZeroQuant-V2 (Yao et al., 2023), Outlier Suppression+ (Wei et al., 2023), FPTQ (Li et al., 2023), QuantEase (Behdin et al., 2023), and OmniQuant (Shao et al., 2023).

7. Conclusions

In summary, the Multilingual Brain Surgeon (MBS) is a groundbreaking approach for improving multilingual LLMs. It tackles the English-centric bias in existing techniques and enhances LLM performance after compressing. Our experiments on the BLOOM model highlight the effectiveness of MBS, benefiting pruning and quantization methods like SparseGPT, Wanda, and GPTQ.

We also studied language interaction during compression, finding that language proportion in the training dataset and language similarity are crucial factors. Languages with larger proportion are less affected by compression, while similar languages perform better when only one language is used in calibration data. Our proposed similarity measure accurately predicts performance drops in such scenarios.

This research not only enhances the practicality of multilingual LLMs compression methods but also maintains language coverage, making multilingual NLP applications more inclusive and powerful.

8. Acknowledgments

This work is funded by the China NSFC Projects (92370206, U23B2057, 62106142 and 62120106006) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

9. Bibliographical References

Kayhan Behdin, Ayan Acharya, Aman Gupta, Sathiya Keerthi, and Rahul Mazumder. 2023. Quantease: Optimization-based quantization for language models – an efficient and intuitive algorithm.

BigScience Workshop. 2022. BLOOM (revision 4ab0472).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler,

Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. 2023. Quip: 2-bit quantization of large language models with guarantees.

Wenhua Cheng, Weiwei Zhang, Haihao Shen, Yiyang Cai, Xin He, and Kaokao Lv. 2023. Optimize weight rounding via signed gradient descent for the quantization of llms.

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

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

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2023. Spqr: A sparse-quantized representation for near-lossless llm weight compression.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Sumanth Doddapaneni, Gowtham Ramesh, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2021. A primer on pretrained multilingual language models.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. 2019. Rigging the lottery: Making all tickets winners. *CoRR*, abs/1911.11134.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers.

Trevor Gale, Erich Elsen, and Sara Hooker. 2019. The state of sparsity in deep neural networks.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. *arXiv preprint arXiv:2105.00572*.

Cong Guo, Jiaming Tang, Weiming Hu, Jingwen Leng, Chen Zhang, Fan Yang, Yunxin Liu, Minyi Guo, and Yuhao Zhu. 2023. OliVe: Accelerating large language models via hardware-friendly outlier-victim pair quantization. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*. ACM.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

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

Itay Hubara, Brian Chmiel, Moshe Island, Ron Banner, Seffi Naor, and Daniel Soudry. 2021. Accelerated sparse neural training: A provable and efficient method to find n:m transposable masks.

Eugenia Iofinova, Alexandra Peste, Mark Kurtz, and Dan Alistarh. 2021. How well do sparse imagenet models transfer? *CoRR*, abs/2111.13445.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. 2023. Squeezellm: Dense-and-sparse quantization.

Mark Kurtz, Justin Kopinsky, Rati Gelashvili, Alexander Matveev, John Carr, Michael Goin, William Leiserson, Sage Moore, Bill Nell, Nir Shavit, and Dan Alistarh. 2020. Inducing and exploiting activation sparsity for fast inference on deep neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5533–5543, Virtual. PMLR.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining.

Yann Le Cun, John S. Denker, and Sara A. Solla. 1989. Optimal brain damage. In *Proceedings*

of the 2nd International Conference on Neural Information Processing Systems, NIPS'89, page 598–605, Cambridge, MA, USA. MIT Press.

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. 2023. Owq: Lessons learned from activation outliers for weight quantization in large language models.

Qingyuan Li, Yifan Zhang, Liang Li, Peng Yao, Bo Zhang, Xiangxiang Chu, Yerui Sun, Li Du, and Yuchen Xie. 2023. Fptq: Fine-grained post-training quantization for large language models.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, Chuang Gan, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration.

Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. 2021. Group fisher pruning for practical network compression. *CoRR*, abs/2108.00708.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models.

Al Mead. 1992. Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 41(1):27–39.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang,

Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Alexandra Peste, Eugenia Iofinova, Adrian Vladu, and Dan Alistarh. 2021. Ac/dc: Alternating compressed/decompressed training of deep neural networks.

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

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2023. Omniquant: Omnidirectionally calibrated quantization for large language models.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023. A simple and effective pruning approach for large language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. Outlier suppression+: Accurate quantization of large language models by equivalent and optimal shifting and scaling.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. 2023. Zeroquant-v2: Exploring post-training quantization in llms from comprehensive study to low rank compensation.

Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. 2023. Rptq: Reorder-based post-training quantization for large language models.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2023. Loraprune: Pruning meets low-rank parameter-efficient fine-tuning.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

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

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*.

10. Language Resource References

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models.

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

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco

Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

A. Details of Calibration Data

Language	Size in Bytes in BLOOM training data	MBS sampling	Equal sampling
en	4.85E+11	87	13
zh-Hans	2.61E+11	47	13
fr	2.08E+11	37	13
es	1.75E+11	31	13
pt	7.93E+10	14	13
ar	7.49E+10	13	13
vi	4.37E+10	7	13
hi	2.46E+10	4	13
id	2.00E+10	3	13
bn	1.86E+10	3	13
ta	7.99E+09	1	13
te	2.99E+09	1	13
ur	2.78E+09	1	13
ne	2.55E+09	1	13
mr	1.78E+09	1	13
gu	1.20E+09	1	13
zh-Hant	7.62E+08	1	12
sw	2.36E+08	1	12
yo	8.97E+07	1	12
ig	1.41E+07	1	12

Table 3: The number of segments taken from each language by each sampling method.

The number of segments taken from each language is detailed in Table 3. We rounded up the segment counts for languages with fewer than one segment to ensure their representation in the calibration data. In the equal sampling scenario, to maintain comparability, some languages have one segment less than others to achieve a total of 256 segments.

B. MBS results tables

Perplexity. Table 6 showcases the perplexity evaluation for each language after pruning on the BLOOM-560m model. The observed trends align closely with those observed on the BLOOM-7b1 model.

Zero shot tasks. Table 7 illustrates the zero-shot task results for the pruned BLOOM-560m model. It is noticeable that the average accuracy using the MBS sampling method continues to outperform the baselines, although the results appear to exhibit more variability. This variability can be attributed to the reduced capacity of smaller models to maintain their multilingual capabilities.

The role of parameter compensation. We have observed a notable distinction in the effects of SparseGPT and Wanda. In monolingual pruning, SparseGPT, which involves parameter updates and employs a more precise pruning metric, appears

Dataset	Dense	Wanda	↑	SparseGPT	↑	MBS + Wanda	↑	MBS + SparseGPT	↑
en	13.68	15.67	15%	14.92	9%	15.55	14%	15.01	10%
zh-Hans	23.70	34.75	47%	35.56	50%	26.59	12%	25.87	9%
fr	9.59	13.16	37%	13.41	40%	10.68	11%	10.39	8%
es	10.71	13.82	29%	13.75	28%	11.91	11%	11.59	8%
pt	10.97	14.58	33%	17.37	58%	12.25	12%	11.96	9%
ar	14.40	29.19	103%	25.33	76%	16.45	14%	15.88	10%
vi	10.16	14.76	45%	15.00	48%	11.59	14%	11.24	11%
hi	10.96	18.26	67%	19.09	74%	12.52	14%	12.19	11%
id	20.48	29.37	43%	37.27	82%	23.76	16%	22.97	12%
bn	17.27	33.37	93%	40.50	134%	20.29	17%	19.51	13%
ta	16.55	42.23	155%	44.10	167%	20.10	21%	19.34	17%
te	18.10	64.97	259%	69.20	282%	24.59	36%	22.05	22%
ur	13.26	27.03	104%	30.56	130%	15.83	19%	15.10	14%
ne	27.22	152.67	461%	148.00	444%	34.90	28%	32.91	21%
mr	23.07	176.78	666%	144.65	527%	32.25	40%	28.91	25%
gu	21.52	184.62	758%	118.48	450%	30.84	43%	26.97	25%
zh-Hant	21.84	113.34	419%	102.26	368%	24.96	14%	24.30	11%
sw	34.35	145.54	324%	135.84	295%	54.32	58%	44.23	29%
yo	53.29	128.12	140%	126.54	137%	79.62	49%	67.52	27%
ig	39.16	90.41	131%	90.89	132%	59.00	51%	49.10	25%
wikitext2	11.37	16.15	42%	13.91	22%	13.82	22%	13.26	17%
Average	20.08	64.70	222%	59.84	198%	**26.28**	**31%**	**23.82**	**19%**

Table 4: Perplexity for each language and their respective increases when compared to the dense BLOOM-7b1 model after pruning. Evaluation performed on XL-Sum and WikiText2 datasets. From top to bottom, languages are ranked in order from the most well-represented to the least represented.

Dataset	Dense	GPTQ	GPTQ+MBS
en	13.68	15.3	15.37
zh-Hans	23.7	28.69	26.28
fr	9.59	10.94	10.46
es	10.71	12.49	11.9
pt	10.97	13.05	12.36
ar	14.4	17.35	16.12
vi	10.16	11.97	11.15
hi	10.96	13.72	12.27
id	20.48	25.36	23.45
bn	17.27	22.88	19.83
ta	16.55	24.5	19.53
te	18.1	32.83	22.67
ur	13.26	18.29	15.4
ne	27.22	45.5	33.97
mr	23.07	43.32	29.57
gu	21.52	40.4	27.83
zh-Hant	21.84	26.85	24.75
sw	34.35	68.42	46.19
yo	53.29	98.46	67.82
ig	39.16	71.56	49.96
wikitext2	11.37	12.6	12.56
Average	20.08	31.17	24.26

Table 5: Perplexity for each language of BLOOM-7b1 model before and after quantization.

to have a more detrimental impact on less well-represented languages. However, when we apply MBS, SparseGPT continues to outperform Wanda. This phenomenon may be attributed to the fact that smaller models are more sensitive to parameter updates. A biased Hessian matrix can exacerbate the model's divergence from the correct direction through these updates. Conversely, a correctly approximated Hessian matrix can effectively guide the pruning in the correct direction.

C. Monolingual pruning results

Distance map of BLOOM-560m model. The distance map depicting the relationships between languages in the BLOOM-560m model is depicted in Figure 8. We can discern a similar clustering pattern to that observed in the BLOOM-7b1 model.

The original distance matrices are provided in the following tables(12, 13).

Dataset	Dense	Wanda	SparseGPT	MBS + Wanda	MBS + SparseGPT
en	13.68	34.30	32.02	35.08	32.87
zh-Hans	23.70	72.92	101.79	62.39	59.28
fr	9.59	22.89	24.99	21.66	20.37
es	10.71	25.41	27.94	23.98	22.56
pt	10.97	27.47	34.99	25.82	24.27
ar	14.40	80.09	1.13E+14	41.36	47.57
vi	10.16	43.26	127.59	30.75	29.64
hi	10.96	44.10	1.20E+15	29.17	43.84
id	20.48	109.13	76038.83	64.40	79.85
bn	17.27	102.78	4.42E+23	56.65	121.11
ta	16.55	176.25	1.71E+07	64.40	209.90
te	18.10	355.98	6.96E+05	116.20	355.54
ur	13.26	97.65	3.62E+20	44.60	61.35
ne	27.22	555.21	1.27E+17	135.67	200.16
mr	23.07	503.51	4.25E+12	142.76	619.28
gu	21.52	327.11	2.17E+13	136.47	197.21
zh-Hant	21.84	137.00	266.49	61.30	58.56
sw	34.35	919.14	6011.36	357.63	336.80
yo	53.29	1.02E+03	3557.88	542.40	450.00
ig	39.16	728.01	1306.30	370.35	307.76
wikitext2	11.37	30.58	29.75	31.09	29.90
Average	20.08	257.99	2.11E+22	114.01	157.51

Table 6: Perplexity for each language and their respective increases when compared to the dense BLOOM-560m model after pruning. Evaluation performed on XL-Sum and WikiText2 datasets. From top to bottom, languages are ranked in order from the most well-represented to the least represented.

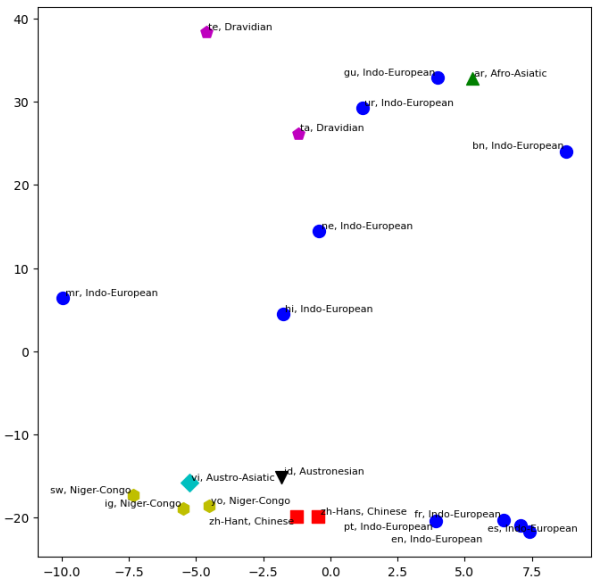

Figure 8: The graph illustrates the relative positions of different languages. Different dot shapes represent different language families. The closer they are on the graph, the more similar they are to each other.

0-shot task	Dense	Wanda	SparseGPT	Wanda equal	SparseGPT equal	MBS+ Wanda	MBS+ SparseGPT
xcopa							
id	59.20%	56.20%	51.80%	58.40%	57.40%	**58.60%**	55.80%
sw	51.60%	52.60%	52.60%	51.80%	**52.80%**	52.20%	52.20%
ta	55.80%	**57.20%**	54.00%	56.00%	56.00%	54.80%	56.20%
vi	61.00%	55.40%	51.60%	**57.60%**	56.20%	56.00%	55.60%
zh	58.60%	52.80%	53.00%	53.40%	53.60%	53.80%	**55.00%**
Average	57.24%	54.84%	52.60%	**55.44%**	55.20%	55.08%	54.96%
xstory_cloze							
ar	52.08%	48.25%	**49.57%**	49.24%	48.31%	48.97%	48.84%
en	61.22%	57.78%	59.23%	56.92%	57.97%	57.51%	**59.70%**
es	55.86%	53.67%	54.40%	53.81%	54.14%	54.27%	**55.00%**
hi	55.00%	52.68%	48.31%	53.21%	**54.00%**	53.08%	52.88%
id	55.53%	53.14%	48.31%	53.14%	52.42%	**53.34%**	52.22%
sw	49.83%	49.11%	48.78%	49.24%	49.24%	**49.44%**	48.51%
te	55.72%	54.33%	53.28%	54.80%	**55.46%**	54.07%	**55.46%**
zh	54.53%	51.95%	51.29%	51.56%	52.68%	51.95%	**53.54%**
Average	54.97%	52.61%	51.65%	52.74%	53.03%	52.83%	**53.27%**
xwinograd							
en	65.89%	61.98%	62.58%	62.28%	**63.44%**	62.02%	62.92%
fr	60.24%	56.63%	51.81%	56.63%	56.63%	**59.04%**	57.83%
pt	60.08%	55.51%	**60.46%**	54.75%	57.41%	55.13%	59.70%
zh	67.66%	66.27%	66.87%	66.87%	63.49%	66.07%	**69.64%**
Average	63.47%	60.10%	60.43%	60.13%	60.24%	60.57%	**62.52%**
pawsx							
en	52.00%	49.90%	48.60%	49.15%	49.85%	47.60%	**50.85%**
es	53.25%	48.75%	48.85%	**51.60%**	48.75%	50.70%	50.80%
fr	47.95%	47.70%	**48.45%**	46.45%	46.75%	46.45%	45.35%
zh	45.20%	45.15%	44.85%	45.50%	**45.70%**	45.55%	44.75%
Average	49.60%	47.88%	47.69%	**48.18%**	47.76%	47.58%	47.94%
xnli							
ar	33.35%	33.47%	33.55%	33.59%	33.57%	33.41%	**33.67%**
en	49.50%	**46.53%**	45.89%	45.43%	46.09%	45.31%	46.37%
es	45.23%	**45.71%**	41.72%	43.11%	42.87%	44.45%	42.95%
fr	45.29%	45.51%	42.20%	45.07%	45.27%	**45.71%**	43.75%
hi	40.84%	38.64%	33.35%	37.90%	36.45%	**38.78%**	34.77%
sw	33.17%	**33.65%**	33.51%	33.45%	33.21%	33.63%	33.29%
ur	37.13%	33.75%	33.27%	**35.23%**	34.77%	35.11%	34.01%
vi	40.52%	**40.28%**	33.05%	38.44%	36.61%	39.14%	37.15%
zh	33.95%	33.33%	33.45%	33.29%	33.47%	33.29%	**33.61%**
Average	39.89%	**38.99%**	36.67%	38.39%	38.04%	38.76%	37.73%
Total Average	51.24%	49.26%	47.95%	49.26%	49.15%	**49.31%**	**49.41%**

Table 7: 0-shot tasks performance on each task of BLOOM-560m pruned model.

Dataset	Dense	en	↑	ig	↑	ta	↑	ur	↑
en	13.68	15.67	15%	17.61	29%	17.20	26%	18.93	38%
zh-Hans	23.70	34.75	47%	33.97	43%	31.99	35%	31.70	34%
fr	9.59	13.16	37%	12.70	32%	13.94	45%	14.84	55%
es	10.71	13.82	29%	14.16	32%	15.83	48%	17.17	60%
pt	10.97	14.58	33%	14.60	33%	15.78	44%	18.20	66%
ar	14.40	29.19	103%	26.27	82%	22.82	58%	17.65	23%
vi	10.16	14.76	45%	12.74	25%	13.48	33%	14.62	44%
hi	10.96	18.26	67%	15.82	44%	13.18	20%	13.97	28%
id	20.48	29.37	43%	27.30	33%	27.48	34%	33.96	66%
bn	17.27	33.37	93%	26.71	55%	21.94	27%	30.11	74%
ta	16.55	42.23	155%	27.14	64%	19.31	17%	24.44	48%
te	18.10	64.97	259%	39.77	120%	25.58	41%	37.68	108%
ur	13.26	27.03	104%	25.07	89%	19.70	49%	15.38	16%
ne	27.22	152.67	461%	64.74	138%	46.59	71%	46.60	71%
mr	23.07	176.78	666%	66.86	190%	45.71	98%	68.97	199%
gu	21.52	184.62	758%	48.30	124%	35.17	63%	37.96	76%
zh-Hant	21.84	113.34	419%	49.41	126%	56.20	157%	36.81	69%
sw	34.35	145.54	324%	58.74	71%	91.94	168%	125.09	264%
yo	53.29	128.12	140%	72.86	37%	127.05	138%	180.01	238%
ig	39.16	90.41	131%	51.30	31%	91.28	133%	149.15	281%
wikitext2	11.37	16.15	42%	16.25	43%	14.14	24%	15.68	38%
Average	20.08	64.70	189.1%	34.40	68.7%	36.49	63.4%	45.19	90.2%

Table 8: Monolingual pruning results using Wanda on BLOOM-7b1 with calibration data in `en`, `ig`, `ta`, and `ur`. Perplexity evaluated on XL-sum and wikitext2. Languages are ranked from the most well-represented to the least represented, from top to bottom.

Dataset	Dense	en		ig		ta		ur	
en	13.68	14.92	9%	16.28	19%	16.53	21%	16.80	23%
zh-Hans	23.70	35.56	50%	30.31	28%	31.45	33%	33.05	39%
fr	9.59	13.41	40%	11.42	19%	12.71	32%	12.75	33%
es	10.71	13.75	28%	12.82	20%	14.12	32%	14.58	36%
pt	10.97	17.37	58%	13.49	23%	14.86	35%	15.38	40%
ar	14.40	25.33	76%	18.54	29%	19.31	34%	16.87	17%
vi	10.16	15.00	48%	12.07	19%	13.26	31%	13.07	29%
hi	10.96	19.09	74%	14.44	32%	12.73	16%	12.46	14%
id	20.48	37.27	82%	25.56	25%	27.34	33%	26.84	31%
bn	17.27	40.50	134%	24.21	40%	21.07	22%	21.85	26%
ta	16.55	44.10	167%	26.36	59%	17.84	8%	22.58	36%
te	18.10	69.20	282%	31.56	74%	24.12	33%	28.66	58%
ur	13.26	30.56	130%	19.20	45%	17.32	31%	14.12	7%
ne	27.22	148.00	444%	47.77	76%	42.62	57%	40.12	47%
mr	23.07	144.65	527%	43.74	90%	37.81	64%	49.41	114%
gu	21.52	118.48	450%	39.39	83%	34.87	62%	32.87	53%
zh-Hant	21.84	102.26	368%	30.51	40%	50.00	129%	43.85	101%
sw	34.35	135.84	295%	46.74	36%	72.81	112%	72.94	112%
yo	53.29	126.54	137%	61.08	15%	105.97	99%	113.09	112%
ig	39.16	90.89	132%	41.37	6%	74.44	90%	86.14	120%
wikitext2	11.37	13.91	22%	13.56	19%	13.66	20%	13.95	23%
Average	20.08	59.84	169.3%	27.64	37.8%	32.14	47.3%	33.40	51.1%

Table 9: Monolingual pruning results using SparseGPT on BLOOM-7b1 with calibration data in `en`, `ig`, `ta`, and `ur`. Perplexity evaluated on XL-sum and wikitext2. Languages are ranked from the most well-represented to the least represented, from top to bottom.

Dataset	Dense	en		ig		te		id	
en	13.68	34.30	151%	49.17	259%	53.12	288%	38.50	181%
zh-Hans	23.70	72.92	208%	87.98	271%	126.56	434%	79.06	234%
fr	9.59	22.89	139%	32.46	238%	41.75	335%	24.91	160%
es	10.71	25.41	137%	38.29	258%	47.38	343%	28.18	163%
pt	10.97	27.47	150%	42.04	283%	57.68	426%	30.47	178%
ar	14.40	80.09	456%	79.26	451%	66.66	363%	55.03	282%
vi	10.16	43.26	326%	39.63	290%	150.26	1379%	38.05	275%
hi	10.96	44.10	303%	50.54	361%	32.37	195%	36.25	231%
id	20.48	109.13	433%	108.26	429%	952.66	4551%	60.77	197%
bn	17.27	102.78	495%	141.25	718%	68.01	294%	83.81	385%
ta	16.55	176.25	965%	147.99	794%	60.52	266%	163.29	887%
te	18.10	355.98	1866%	260.31	1338%	95.77	429%	445.67	2362%
ur	13.26	97.65	636%	143.11	979%	49.14	271%	63.39	378%
ne	27.22	555.21	1940%	320.61	1078%	143.19	426%	237.41	772%
mr	23.07	503.51	2083%	290.52	1159%	140.02	507%	220.62	856%
gu	21.52	327.11	1420%	215.13	900%	124.55	479%	286.66	1232%
zh-Hant	21.84	137.00	527%	156.15	615%	227.56	942%	167.85	668%
sw	34.35	919.14	2576%	419.52	1121%	897.07	2511%	465.66	1256%
yo	53.29	1024.94	1823%	529.35	893%	711.20	1235%	573.17	976%
ig	39.16	728.01	1759%	285.28	629%	773.60	1876%	409.11	945%
wikitext2	11.37	30.58	169%	40.70	258%	46.07	305%	33.40	194%
Average	20.08	257.99	883.9%	165.60	634.4%	231.67	850.2%	168.63	610.0%

Table 10: Monolingual pruning results of Wanda on BLOOM-560m.

Dataset	Dense	en	ig	te	id
en	14	32	43	48	40
zh-Hans	24	102	128	108	110
zh-Hant	22	266	162	227	279
fr	10	25	28	39	28
es	11	28	34	48	32
pt	11	35	37	54	35
ar	14	1.13E+14	8.10E+07	247	3.52E+07
vi	10	128	70	948	78
hi	11	1.20E+15	3.93E+10	128	2.98E+09
id	20	76039	7049	11397	62
bn	17	4.42E+23	2.16E+10	1788	6.12E+15
ta	17	17114940	361068	85	1.80E+08
te	18	696319	180476	75	2.40E+07
ur	13	3.62E+20	55283	807	1.55E+07
ne	27	1.27E+17	1.70E+09	636	6.43E+10
mr	23	4.25E+12	2.13E+08	1250	3.76E+11
gu	22	2.17E+13	3051	252	4.96E+06
sw	34	6011	467	1363	1502
yo	53	3558	342	882	1463
ig	39	1306	205	990	753
wikitext2	11	30	36	41	34
Average	20	2.11E+22	2.99E+09	1020	2.92E+14

Table 11: Monolingual pruning results of SparseGPT on BLOOM-560m.

	en	zh-Hans	zh-Hant	fr	es	pt	ar	vi	hi	id	bn	ta	te	ur	ne	mr	gu	sw	yo	ig
en	0	10	11	5	6	7	9	8	10	6	9	8	12	20	14	9	12	10	10	10
zh-Hans	10	0	1	13	15	16	8	12	3	12	6	6	5	11	8	5	6	16	17	18
zh-Hant	11	1	0	15	17	18	8	14	4	13	6	6	5	10	7	6	6	17	19	19
fr	5	13	15	0	2	3	11	5	12	4	10	10	14	22	16	11	13	6	6	7
es	6	15	17	2	0	1	12	7	15	6	13	12	17	25	18	13	15	7	5	6
pt	7	16	18	3	1	0	12	7	15	5	13	13	17	25	18	13	16	5	4	5
ar	9	8	8	11	12	12	0	8	6	8	6	5	8	14	8	4	8	11	12	12
vi	8	12	14	5	7	7	8	0	10	2	8	8	12	19	12	8	10	4	6	6
hi	10	3	4	12	15	15	6	10	0	10	3	3	3	10	5	3	4	14	16	16
id	6	12	13	4	6	5	8	2	10	0	8	7	12	19	13	8	10	4	6	6
bn	9	6	6	10	13	13	6	8	3	8	0	2	4	12	6	3	3	12	14	14
ta	8	6	6	10	12	13	5	8	3	7	2	0	4	12	6	2	4	11	13	13
te	12	5	5	14	17	17	8	12	3	12	4	4	0	8	4	5	2	16	18	18
ur	20	11	10	22	25	25	14	19	10	19	12	12	8	0	7	12	10	23	25	25
ne	14	8	7	16	18	18	8	12	5	13	6	6	4	7	0	5	5	16	18	18
mr	9	5	6	11	13	13	4	8	3	8	3	2	5	12	5	0	4	12	13	13
gu	12	6	6	13	15	16	8	10	4	10	3	4	2	10	5	4	0	14	16	17
sw	10	16	17	6	7	5	11	4	14	4	12	11	16	23	16	12	14	0	3	3
yo	10	17	19	6	5	4	12	6	16	6	14	13	18	25	18	13	16	3	0	1
ig	10	18	19	7	6	5	12	6	16	6	14	13	18	25	18	13	17	3	1	0
Average	9.3	9.4	10.1	9.25	10.6	10.65	8.5	8.3	8.1	7.95	7.6	7.25	9.2	15.45	10.2	7.45	8.75	10.2	11.1	11.35

Table 12: Original distance matrix generated from the BLOOM-7b1 model.

	en	zh-Hans	zh-Hant	fr	es	pt	ar	vi	hi	id	bn	ta	te	ur	ne	mr	gu	sw	yo	ig
en	0	8	9	2	1	4	54	14	28	12	47	48	60	52	38	31	55	15	13	13
zh-Hans	8	0	1	7	8	5	53	7	24	6	45	46	58	49	34	28	53	7	5	5
zh-Hant	9	1	0	8	9	6	53	6	24	5	45	46	58	49	34	28	53	6	5	5
fr	2	7	8	0	2	4	53	12	26	10	46	47	59	51	36	30	53	13	12	12
es	1	8	9	2	0	3	54	13	27	11	46	47	59	51	37	31	54	14	12	13
pt	4	5	6	4	3	0	53	11	26	9	45	46	59	50	36	29	53	11	9	9
ar	54	53	53	53	54	53	0	50	29	49	12	7	9	7	19	30	5	51	52	52
vi	14	7	6	12	13	11	50	0	22	5	42	43	55	46	31	26	50	5	6	6
hi	28	24	24	26	27	26	29	22	0	20	23	22	34	25	11	9	29	23	23	24
id	12	6	5	10	11	9	49	5	20	0	41	42	54	45	30	23	49	6	6	5
bn	47	45	45	46	46	45	12	42	23	41	0	10	20	9	14	26	9	43	43	44
ta	48	46	46	47	47	46	7	43	22	42	10	0	13	6	13	22	9	44	45	45
te	60	58	58	59	59	59	9	55	34	54	20	13	0	13	25	32	12	56	57	57
ur	52	49	49	51	51	50	7	46	25	45	9	6	13	0	15	26	6	47	48	48
ne	38	34	34	36	37	36	19	31	11	30	14	13	25	15	0	13	20	32	33	33
mr	31	28	28	30	31	29	30	26	9	23	26	22	32	26	13	0	31	26	26	26
gu	55	53	53	53	54	53	5	50	29	49	9	9	12	6	20	31	0	51	51	52
sw	15	7	6	13	14	11	51	5	23	6	43	44	56	47	32	26	51	0	5	4
yo	13	5	5	12	12	9	52	6	23	6	43	45	57	48	33	26	51	5	0	3
ig	13	5	5	12	13	9	52	6	24	5	44	45	57	48	33	26	52	4	3	0
Average	25.2	22.45	22.5	24.15	24.6	23.4	34.6	22.5	22.45	21.4	30.5	30.05	39.5	32.15	25.2	24.65	34.75	22.95	22.7	22.8

Table 13: Original distance matrix generated from the BLOOM-560m model.