Language-Specific Pruning for Efficient Reduction of Large Language Models

Maksym Shamrai
Institute of Mathematics of NAS of Ukraine
Kyiv Academic University
Kyiv, Ukraine
m.shamrai@imath.kiev.ua

Abstract

Delving into pruning techniques is essential to boost the efficiency of Large Language Models (LLMs) by reducing their size and computational demands, resulting in faster and more cost-effective inference. In this work, our key contribution lies in recognizing that LLMs trained on diverse languages manifest distinct language-specific weight distributions. Exploiting this insight, we illustrate that pruning LLMs using language-specific data results in a more potent model compression. Empirical evidence underscores the critical nature of pruning on language-specific data, highlighting a noteworthy impact on the perplexity of Ukrainian texts compared to pruning on English data. The proposed methodology significantly reduces the size of LLaMA, LLaMA 2 and Mistral models while preserving competitive performance. This research underscores the significance of linguistic considerations in LLM pruning and advocates for language-specific optimization, establishing a framework for more efficient and tailored language models across diverse linguistic contexts. Additionally, all experiments were conducted using a single consumer-grade NVIDIA RTX 3090 GPU, and the code is available at https://github.com/mshamrai/language-specific-pruning.

Keywords: Language Model Pruning, Large Language Models, Language-Specific Optimization, Ukrainian Language Processing

1. Introduction

The evolution of Large Language Models (LLMs) has unlocked unprecedented capabilities in natural language processing, yet the monumental size of these models necessitates innovative solutions for their efficient deployment. Lately, quantization techniques, which employ lower precision types for compression, have enhanced the accessibility of LLMs to a broader audience (Frantar et al., 2022; Dettmers et al., 2022, 2024). While these advancements are noteworthy, alternative compression methods can yield significant improvements. Pruning, a technique involving the selective removal of model weights, is a promising avenue for addressing computational challenges without compromising performance.

While existing pruning methods have demonstrated success in general contexts (Molchanov et al., 2019; Yang et al., 2022; Ma et al., 2024), their application to different languages and the implications for model performance remain largely unexplored. This paper pioneers the investigation of language-specific pruning for LLMs, with a dedicated focus on the Ukrainian language. Our objective is to establish that the efficacy of pruning methods is linked to the linguistic characteristics of the target language. Leveraging state-of-the-art techniques such as SparseGPT (Frantar and Alistarh, 2023) and Wanda (Sun et al., 2023), our method achieves competitive perplexity scores when evaluated on a Ukrainian dataset with sparse versions of LLaMA (Touvron et al., 2023a), LLaMA 2 (Touvron et al., 2023b) and Mistral (Jiang et al., 2023) models, eliminating the necessity for retraining.

Moreover, considering that pruning strategies in Transformer-based models primarily target linear layers due to their significant presence and crucial role in model parameterization, the methods employed and our findings are applicable to any Transformer architecture without constraints.

It is essential to note that the successful application of SparseGPT and Wanda requires reference data to tailor the pruning specifically for the characteristics of the given dataset. For our Ukrainian language exploration, we utilized reference data sourced from UberText 2.0 (Chaplynskyi, 2023) – this corpus provides a robust foundation for assessing the effectiveness of language-specific pruning in real-world linguistic contexts.

Additionally, we delve into the ramifications of language-specific pruning on model performance. To emphasize the language-specific nature of our findings, we conducted additional experiments by attempting to prune models on the English c4 dataset (Raffel et al., 2019).

The evaluation of pruning methods for the Ukrainian language includes a comparison of perplexity metrics for dense, unstructured, and 2:4 semi-structured sparsity patterns with 50% sparsity, indicating a pruning of models by half. The adoption of a 2:4 semi-structured sparsity pattern, where at least two out of every four elements must be zero, is investigated due to its native support in the NVIDIA Ampere GPU architecture, leading to significant computational speed-ups (Mishra et al.,

The Third Ukrainian Natural Language Processing Workshop (UNLP) @LREC-COLING-2024, pages 135–140
25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

2021).

In conclusion, this research marks a pioneering effort in advancing the efficiency of Large Language Models (LLMs) through the exploration of language-specific pruning techniques, with a focused examination on the Ukrainian language. Our primary contribution lies in establishing a profound connection between the efficacy of pruning methods and the unique linguistic characteristics of the target language.

2. Related work

While our work primarily focuses on training-free approaches to language model pruning, it is essential to acknowledge the existence of methods that require post-pruning retraining (Jiao et al., 2019; Ma et al., 2024). The effectiveness of such methods is contingent on the availability and quality of training data, making it less practical for scenarios where acquiring sufficient annotated data is a formidable task.

In the context of low-resource languages such as Ukrainian, where limited annotated data poses a significant obstacle, this limitation underscores the importance of investigating training-free approaches, which mitigate the need for additional labeled data. Therefore, we focus on the methods that requires only a relatively small calibration dataset for efficient model pruning.

These approaches share a similar concept: assessing weight importance based on a specific metric and input calibration data, where a larger value of the importance metric indicates that the weight should be retained. The pruning process is conducted in a layer-wise manner, involving the calculation of weight importance for each layer. Subsequently, the weights are sorted, and depending on the desired sparsity level, weights with lower importance are replaced with zeros. This streamlined approach facilitates efficient pruning, even for large-scale models.

The subsequent subsections delve into the details of this methods, highlighting its potential and practicality in the context of low-resource languages.

2.1. SparseGPT

In recent strides towards optimizing the efficiency of Large Language Models (LLMs), SparseGPT emerges as a pioneering one-shot pruning method (Frantar and Alistarh, 2023).

The foundation of SparseGPT's pruning methodology lies in the formalization of the problem through a local layer-wise reconstruction approach. It employs a pruning metric that considers the layer-wise reconstruction problem.

$$\mathbf{S}_{ij} = \left[|\mathbf{W}|^{\mathbf{2}} / \text{diag}\left((\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda \mathbf{I})^{-\mathbf{1}} \right) \right]_{\mathbf{ij}} \quad (1)$$

The weight importance metric utilized in SparseGPT, represented by Equation 1, incorporates the Hessian matrix in the denominator, where $\mathbf{W}$ denotes the weights, $\mathbf{X}$ represents the inputs, and λ stands for the Hessian dampening factor, employed to prevent the collapse of inverse computation. This metric underscores the importance of local layer-wise information during the pruning process. By prioritizing such information, SparseGPT ensures the preservation of accuracy levels crucial for the optimal performance of large language models.

2.2. Wanda

The approach, termed "Pruning by Weights and Activations" (Sun et al., 2023) presents an effective solution to the pruning challenge. Wanda augments the standard weight magnitude pruning metric with input activations, effectively evaluating weight importance.

$$\mathbf{S}_{ij} = |\mathbf{W}_{\mathbf{ij}}| \cdot ||\mathbf{X}_{\mathbf{ij}}||_2 \quad (2)$$

The computation of weight importance in Wanda is defined by Equation 2, where the score for each individual weight $\mathbf{W}_{\mathbf{ij}}$ is computed as the product of its magnitude and the corresponding norm of input feature $\mathbf{X}_{\mathbf{ij}}$. Therefore, the score encapsulates the weight's importance within the context of its associated input activations.

One of the key strengths of Wanda lies in its computational efficiency and minimal memory overhead. The method can be executed in a single forward pass, making it suitable for practical implementation in large-scale language models.

In summary, SparseGPT and Wanda employ different weight importance metrics, each grounded in a common conceptual framework. While SparseGPT utilizes a more complex metric, Wanda prioritizes computational efficiency. Following sections will explore comparative analyses to assess the effectiveness of each method for language-specific pruning.

3. Experimental Methodology and Setup

In this section detailing our experimental methodology and setup for the pruning experiments, we chose models from the LLaMA and Mistral families, specifically opting for LLaMA 7B, LLaMA 2 7B and Mistral v0.1 7B in 16-bit floating point precision.

To evaluate the models, we utilize the perplexity metric, which measures the effectiveness of a language model in predicting a sequence. Perplexity is computed as the exponentiated average negative log-likelihood of a sequence, representing the level of surprise or uncertainty of the model in predicting the next token. Mathematically, if we have a tokenized sequence $X = (x_0, x_1, \ldots, x_t)$, then the perplexity of X is calculated using the equation:

$$\text{PPL}(X) = exp\{-\frac{1}{t}\sum_{i=0}^{t}\log p_\theta(x_i|x_{<i})\},$$

where $\log p_\theta(x_i|x_{<i})$ denotes the log-likelihood of the ith token conditioned on the preceding tokens $x_{<i}$, according to our model parameterized by θ. Therefore, a higher value of perplexity indicates poorer predictions, while lower perplexity values signify better model performance.

Our focus on pruning and subsequent evaluation centered around the Ukrainian language, and for this, we utilized the UberText 2.0 corpus (Chaplynskyi, 2023), encompassing various subcorpora such as court, fiction, news, and Wikipedia. Excluding the social subcorpus, which predominantly contains short texts, we randomly selected 1000 samples for calibration and 50 samples for evaluation from each relevant subcorpus. In total, the calibration dataset consisted of 4000 samples, while the evaluation dataset consisted of 200 samples. These selections contributed to the creation of robust calibration and evaluation datasets, with each sample exceeding a length of 8192 characters.

To calibrate the model effectively, we implemented a random sampling approach from the calibration dataset, utilizing a specified seed along with the number of calibration samples as input arguments. The evaluation process covered the full evaluation dataset, calculating perplexity. Experiments were conducted with varying numbers of calibration samples and three distinct seeds to ensure statistical robustness, with mean and standard deviation calculations performed across multiple runs involving different seeds.

To underscore the importance of linguistic considerations, we expanded our experimentation to include the pruning of models on the c4 dataset, written in English. The subsequent evaluation was carried out on the Ukrainian-language evaluation dataset. Furthermore, to comprehensively assess and compare pruning performance, we also evaluated the dense version of the models (i.e., the original models without pruning) on the same dataset.

Our experiments included the introduction of diverse sparsity structures, such as unstructured and semi-structured 2:4 sparsity. Each configuration aimed to achieve a 50% sparsity level, indicating that half of the weights in each linear layer were pruned.

Overall, the objective of the experiments is to empirically and statistically investigate several key aspects:

1. The impact of the size of the calibration dataset on the performance of pruned models.
2. Comparison of different pruning methods to determine their efficacy for language-specific tasks.
3. Assessment of the significance of the language used in the calibration data for pruning effectiveness.

These experiments aim to provide insights into the factors influencing model performance post-pruning, identify optimal pruning methods tailored to language-specific requirements, and ascertain the relevance of language-specific calibration data for pruning outcomes.

Regarding the hardware requirements of the methods, both are capable of pruning 7B models in a matter of hour on a single NVIDIA RTX 3090. Pruning larger-scale models is also feasible but requires additional computational resources. For instance, in a study by Frantar and Alistarh (2023), the authors demonstrate that their method can prune a 175B model on a single NVIDIA A100 GPU. Overall, based on the experiments conducted, we can conclude that the pruning requirements primarily depend on the size of the model and its contextual window, without incurring additional overhead. Therefore, the pruning requirements are approximately equivalent to those of inference.

4. Results

In this section, we present and discuss the outcomes of our experiments, focusing on the perplexity metric evaluated on the Ukrainian evaluation dataset with various setups for different models.

Table 1 illustrates perplexity values for models pruned on UberText 2.0 dataset, employing both unstructured and 2:4 semi-structured pruning configurations with 50% sparsity. Additionally, the models underwent pruning using diverse calibration sample sizes (64, 128, 256, 512) to examine the relationship between sample size and performance.

Analyzing the table, it could be observed that Wanda's performance appears independent of calibration set size or, perhaps, this correlation does not consistently hold across all models. This is particularly evident in the perplexity values of unstructured models, such as Mistral v0.1 7B, where the Pearson correlation between calibration set size and perplexity mean values is 0.99, and LLaMA 2 7B, where the correlation is -0.98. Conversely, all

Method	Calibration Samples	LLaMA 7B	LLaMA 2 7B	Mistral v0.1 7B
Unstructured Wanda	64	12.162 ± 0.025	11.283 ± 0.007	**9.314 ± 0.098**
	128	12.161 ± 0.012	11.278 ± 0.007	9.726 ± 0.125
	256	**12.148 ± 0.008**	11.275 ± 0.009	10.385 ± 0.038
	512	12.152 ± 0.007	**11.254 ± 0.012**	12.262 ± 0.424
2:4 Wanda	64	31.533 ± 0.169	**30.101 ± 0.406**	**29.822 ± 0.381**
	128	31.438 ± 0.348	30.177 ± 0.361	30.741 ± 0.231
	256	31.496 ± 0.327	30.651 ± 0.353	32.709 ± 0.328
	512	**31.198 ± 0.446**	30.883 ± 0.271	34.471 ± 0.704
Unstructured SparseGPT	64	10.632 ± 0.027	9.703 ± 0.013	7.109 ± 0.003
	128	10.559 ± 0.011	9.683 ± 0.028	7.095 ± 0.011
	256	10.531 ± 0.006	9.671 ± 0.015	7.085 ± 0.003
	512	**10.529 ± 0.020**	**9.652 ± 0.012**	**7.074 ± 0.004**
2:4 SparseGPT	64	13.319 ± 0.092	11.559 ± 0.082	8.582 ± 0.036
	128	13.148 ± 0.192	11.515 ± 0.072	8.551 ± 0.041
	256	13.093 ± 0.054	11.457 ± 0.035	8.497 ± 0.006
	512	**12.994 ± 0.047**	**11.379 ± 0.008**	**8.476 ± 0.031**

Table 1: Perplexity values of different models and different pruning configuration.

models pruned by SparseGPT exhibit a notably high negative correlation, such as for 2:4 LLaMA 7B, where the correlation is -0.9. Hence, we can assert that Wanda's performance is not necessarily dependent on the calibration data size, while SparseGPT's performance does show such dependency. This difference could be attributed to the inherent dissimilarity in the precision of importance metrics employed by each method, where Wanda utilizes a faster but less accurate metric, and SparseGPT employs a more precise but time-intensive alternative.

The Table 3 presents the optimal perplexity values achieved by models pruned using both unstructured and 2:4 semi-structured configurations, each with 50% sparsity, on calibration data from UberText 2.0 or c4 datasets. Additionally, the perplexity values for the dense models are included.

The analysis of the table leads to the conclusion that, among both unstructured and 2:4 semi-structured configurations, the most effective pruning method is SparseGPT when applied to the UberText 2.0 dataset, which consists of Ukrainian texts. It is also noteworthy that the superiority of the SparseGPT pruning technique becomes evident, particularly when the pruning pattern is 2:4 semi-structured.

Furthermore, the extreme variances observed in models pruned with c4 data indicate a significant dependency on randomness in the pruning process, suggesting that the outcome is less influenced by the dataset itself.

Moreover, we analyze the memory footprint of the models before and after pruning. As shown in Table 2, pruning with a 50% sparsity level reduces the memory size of the models by approximately 41%. Therefore, pruning enables a significant decrease in the memory consumption of the model's parameters while preserving parameters in 16-bit floating-point format. However, achieving such a reduction in memory usage is not feasible with unstructured sparsity. To attain this reduction, we should utilize a 2:4 semi-structured sparsity pattern, which employs an efficient sparse semi-structured tensor representation.

Model	Dense	Sparse
LLaMA 7B	12.58 Gbs	7.31 Gbs
LLaMA 2 7B	12.68 Gbs	7.40 Gbs
Mistral v0.1 7B	13.99 Gbs	8.30 Gbs

Table 2: Memory footprint before (dense) and after (sparse) pruning with 50% sparsity level and 2:4 semi-structured sparsity configuration of different models.

Additionally, among these three models, Mistral v0.1 7B demonstrates the best pruning performance, as indicated by the lowest residual between dense and pruned perplexity values.

Therefore, SparseGPT emerges as the preferred pruning method for language-specific applications, with its performance significantly influenced by the language of the calibration dataset.

5. Conclusion

In this study, we conducted a comprehensive set of experiments to investigate the impact of pruning methodologies on language models, with a specific

Model	LLaMA 7B	LLaMA 2 7B	Mistral v0.1 7B
Dense	8.950	8.269	6.460
Unstructured Wanda on c4	13.953 ± 0.060	13.829 ± 0.087	41.466 ± 6.314
Unstructured SparseGPT on c4	15.797 ± 0.761	15.011 ± 0.283	9.208 ± 0.086
Unstructured Wanda on UberText 2.0	12.148 ± 0.008	11.254 ± 0.012	9.314 ± 0.098
Unstructured SparseGPT on UberText 2.0	**10.529 ± 0.020**	**9.652 ± 0.012**	**7.074 ± 0.004**
2:4 Wanda on c4	52.346 ± 1.628	79.801 ± 7.338	433.940 ± 282.154
2:4 SparseGPT on c4	89.772 ± 28.306	57.460 ± 5.379	165.516 ± 90.769
2:4 Wanda on UberText 2.0	31.198 ± 0.446	30.101 ± 0.406	29.822 ± 0.381
2:4 SparseGPT on UberText 2.0	**12.994 ± 0.047**	**11.379 ± 0.008**	**8.476 ± 0.031**

Table 3: Perplexity values of different models and different pruning configuration.

focus on language-specific considerations. Our objectives were in the following:

1. **Dependency on Calibration Dataset Size:**

   The experiments aimed to state whether the performance of pruned models is influenced by the size of the calibration dataset. Results revealed that, unlike SparseGPT, the Wanda pruning method demonstrated little to no dependence on the calibration set size.

2. **Comparison of the Pruning Methods:**

   Through an analysis of perplexity values, we compared two language-specific pruning methods, Wanda and SparseGPT. The latter emerged as the preferred pruning method for language-specific applications, particularly under 2:4 semi-structured pruning configurations.

3. **Language Dependence in Pruning Performance:**

   Our investigation extended to clarify whether the pruning methods yield distinct outcomes based on the language of the calibration dataset. The results clearly demonstrated that the effectiveness significantly dependent on the language of the calibration data.

Our findings contribute valuable insights into the language-specific considerations of model pruning, paving the way for more informed choices in deploying such techniques for diverse natural language processing applications.

6. Discussion and Future Work

Our experiments reveal that different sets of parameters are optimal for different languages. In particular, an LLM pruned on English calibration data shows lower performance on the Ukrainian evaluation dataset compared to an LLM pruned on Ukrainian calibration data. Consequently, this pruning technique can serve as a foundational framework for linguistic comparisons among languages. For instance, a compelling exploration could involve comparing the languages of Polish and Ukrainian, given their Slavic roots and linguistic proximity. Demonstrating their linguistic closeness in the LLM context suggests that fine-tuning the LLM on data from both languages could potentially enhance overall performance.

Furthermore, it's essential to assess alternative training-free pruning techniques, such as those proposed by Zhang et al. (2023), to conduct a comprehensive investigation before developing a truly innovative, language-specific pruning approach.

In addition, the next phase of research could explore the synergies between pruning and quantization, aiming to create the smallest and fastest Ukrainian LLM. Combining these techniques holds the promise of optimizing model size and inference speed, contributing to more efficient language models.

7. References

Dmytro Chaplynskyi. 2023. Introducing UberText 2.0: A corpus of modern Ukrainian at scale. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop*, pages 1–10, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accu-

rately pruned in one-shot. *arXiv preprint arXiv:2301.00774*.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2024. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36.

Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. 2021. Accelerating sparse deep neural networks. *arXiv preprint arXiv:2104.08378*.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272.

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*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Nakyeong Yang, Yunah Jang, Hwanhee Lee, Seohyeong Jung, and Kyomin Jung. 2022. Attribution-based task-specific pruning for multi-task language models. *arXiv preprint arXiv:2205.04157*.

Yuxin Zhang, Lirui Zhao, Mingbao Lin, Yunyun Sun, Yiwu Yao, Xingjia Han, Jared Tanner, Shiwei Liu, and Rongrong Ji. 2023. Dynamic sparse no training: Training-free fine-tuning for sparse llms. *arXiv preprint arXiv:2310.08915*.