Vicomtech@WMT 2025: Evolutionary Model Compression for Machine Translation

David Ponce[1,2] and **Harritxu Gete**[1] and **Thierry Etchegoyhen**[1]
[1] Fundación Vicomtech, Basque Research and Technology Alliance (BRTA)
[2] University of the Basque Country EHU
{adponce,hgete,tetchegoyhen}@vicomtech.org

Abstract

We describe Vicomtech's participation in the WMT 2025 Shared Task on Model Compression. We addressed all three language pairs of the constrained task, namely Czech to German, English to Arabic and Japanese to Chinese, using the Aya Expanse 8B model as our base model. Our approach centers on GeLaCo, an evolutionary method for LLM compression via layer collapse operations, which efficiently explores the compression solution space through population-based search and a module-wise similarity fitness function that captures attention, feed-forward, and hidden state representations. We systematically evaluated compression at three different ratios (0.25, 0.50, and 0.75) and applied targeted post-training techniques to recover performance through fine-tuning and knowledge distillation over translation instructions. Additionally, we explored quantization techniques to achieve further model size reduction. Our experimental results demonstrate that the combination of evolutionary layer compression, targeted post-training, and quantization can achieve substantial model size reduction while maintaining competitive translation quality across all language pairs.

1 Introduction

The remarkable success of Large Language Models (LLMs) across diverse natural language processing tasks (Radford et al., 2019; Brown et al., 2020; Chang et al., 2024) has established them as powerful tools for language understanding and generation. Beyond their general capabilities, LLMs have also demonstrated remarkable effectiveness in machine translation tasks, often matching or exceeding the performance of dedicated neural machine translation systems (Xu et al., 2024; Zhu et al., 2024a; Kocmi et al., 2023, 2024). Simultaneously, recent work has focused on the development of specialized multilingual LLMs designed specifically for translation and cross-lingual tasks, such as Aya Expanse (Dang et al., 2024), EuroLLM (Martins et al., 2025), and Tower (Alves et al., 2024).

However, these advances come at the cost of substantial computational requirements. Modern LLMs, ranging from billions to trillions of parameters, demand considerable memory and processing power for both training and inference, with associated environmental impacts that raise serious sustainability concerns (Strubell et al., 2019). These computational requirements create barriers to widespread deployment and usage where reduced memory footprint and efficient inference are essential for practical adoption.

In this work, we describe Vicomtech's participation in the constrained track of the WMT 2025 Model Compression shared task (Gaido et al., 2025). This task focuses specifically on making LLMs suitable for deployment in machine translation within resource-constrained environments. The task evaluates compression techniques across multiple dimensions: model size reduction, translation quality preservation, and inference speed optimization. Participants were tasked to compress the Aya Expanse 8B model while maintaining competitive translation performance across three language pairs: Czech-German, English-Arabic, and Japanese-Chinese.

To address these challenges, we employed GeLaCo (Ponce et al., 2025), an evolutionary algorithm for LLM compression that builds upon the layer collapse operations of LaCo (Yang et al., 2024). GeLaCo efficiently explores the compression solution space through population-based search and a module-wise similarity fitness function that captures attention, feed-forward, and hidden state representations. We systematically applied this approach across multiple compression

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 769–779
November 8–9, 2025.

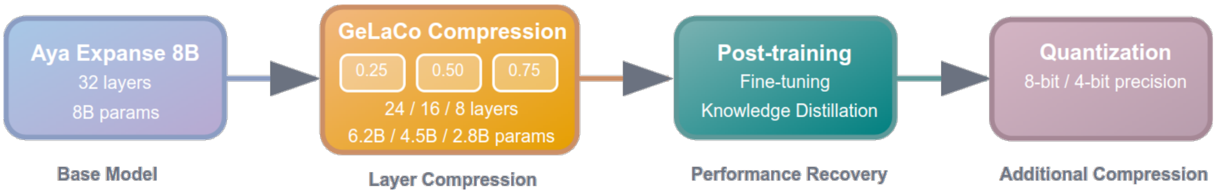

Figure 1: Overview of our model compression pipeline. Starting from Aya Expanse 8B (32 layers, 8B parameters), we apply GeLaCo compression at three ratios (0.25, 0.50, 0.75), reducing to 24, 16, or 8 layers respectively. Compressed models undergo post-training via Supervised Fine-Tuning or Generalized Knowledge Distillation for performance recovery, followed by optional 8-bit or 4-bit quantization for additional size reduction.

ratios (0.25, 0.50, and 0.75) for all three language pairs. We explored targeted post-training techniques, including continued pre-training and knowledge distillation, to recover translation performance after compression. Additionally, we used quantization methods to achieve further size reduction while maintaining translation quality.

Our experimental results demonstrate that the combination of evolutionary layer compression, targeted post-training, and quantization can achieve substantial model size reduction while preserving competitive translation capabilities across diverse language pairs. Figure 1 presents an overview of our compression pipeline.

2 Background

Model Compression. Traditional compression techniques for large language models include quantization, knowledge distillation, and pruning. Among pruning approaches, structured methods that remove entire layers or components have shown particular promise. Notable recent methods include SliceGPT (Ashkboos et al., 2024), which replaces sparse weight matrices with smaller dense matrices; LLM-Pruner (Ma et al., 2023), which uses gradient information to identify prunable components; and LaCo (Yang et al., 2024), which merges layers based on cosine similarity differences. However, these approaches typically require costly empirical evaluation of different compression schemes.

Evolutionary Compression. Evolutionary algorithms have recently emerged as a principled approach to explore the compression solution space. EvoPress (Sieberling et al., 2024) formulates compression as a general optimization problem using evolutionary algorithms for dynamic, non-uniform compression. DarwinLM (Tang et al., 2025) introduces training-aware structured pruning within an evolutionary framework. These methods demonstrate the potential of evolutionary approaches to discover better compression configurations compared to heuristic methods.

LLMs for Machine Translation. Large language models have become the dominant paradigm in machine translation, often matching or exceeding dedicated neural MT systems (Kocmi et al., 2023, 2024). This shift has motivated the development of specialized multilingual LLMs for translation tasks, such as Aya Expanse (Dang et al., 2024), which serves as the base model for our compression experiments. The success of LLMs in translation, combined with their substantial computational requirements, makes efficient compression particularly important for practical deployment in translation scenarios.

3 Methodology

3.1 GeLaCo

We employed GeLaCo, an evolutionary algorithm for LLM compression based on layer collapse operations. Layer collapse reduces model size by merging consecutive layers through differential weight merging, where the resulting parameters when merging m consecutive layers starting from layer l are computed as in Equation 1:

$$\theta_l^* = \theta_l + \sum_{k=1}^{m} (\theta_{l+k} - \theta_l), \tag{1}$$

where θ_l denotes the weight parameters of layer l, and $(\theta_{l+k} - \theta_l)$ denotes the parameter difference between each subsequent layer and the base layer l. This preserves contributions from collapsed layers while reducing the overall model size.

The main challenge in layer collapse lies in determining optimal merge operation combinations, as the search space grows exponentially with model size. Other approaches such as LaCo rely on empirical evaluation using heuristic methods, which can be computationally expensive and may miss better compression solutions due to a limited exploration of the solution space.

GeLaCo addresses these limitations via population-based evolutionary search that efficiently explores compression configurations. The method uses a module-wise similarity fitness function that captures attention, feed-forward, and hidden state representations to guide the layer collapse operations through differential weight merging. The evolutionary process maintains a population of candidate solutions, where each individual represents a specific configuration of layer merge operations, evolving through fitness evaluation, selection, and crossover operations.

The fitness function evaluates compressed model quality by computing module-wise similarity between the original and compressed models using a small calibration dataset. For each calibration sentence, GeLaCo calculates cosine similarity across attention modules, feed-forward network components, and final hidden state representations, with the overall fitness score averaged across all three components and all calibration sentences. This approach enables an efficient evaluation of compression quality during the evolutionary search using only a small set of representative text samples.

3.2 Post-training

Previous work has demonstrated that instruction-following capabilities can be partially recovered through post-training of compressed models (Chen et al., 2025; Men et al., 2025; Ponce et al., 2025). We explored two distinct approaches for performance recovery. First, we applied Supervised Fine-Tuning (SFT) over translation instructions, where we adapted the compressed models to the translation task through continued training on parallel data. Alternatively, we explored Generalized Knowledge Distillation (GKD) (Agarwal et al., 2024), which addresses distribution mismatch by training the compressed student model on its own generated sequences while leveraging feedback from the original teacher model.

3.3 Quantization

To achieve further compression beyond layer collapse, we explored quantization techniques as a complementary approach. Quantization (Gray and Neuhoff, 1998) reduces the numerical precision of model parameters, offering additional size reductions while maintaining competitive performance (Zhu et al., 2024b). We systematically evaluated the combined effects of layer compression and quantization to understand their complementary potential for model size reduction.

3.4 In-context Learning

We investigated the effectiveness of in-context learning (ICL) to enhance the performance of compressed models across different prompting strategies. We explored three distinct setups: zero-shot translation, where we provided no examples; static few-shot learning, using a fixed set of translation examples; and retrieval augmented generation (RAG), using a dynamic similarity-based retrieval where we selected examples based on their relevance to the input sentence. This analysis allowed us to understand how compressed models respond to different contextual information and whether in-context learning can compensate for performance degradation from compression.

Dataset name	**Total Size**	**Samples**
CES-DEU		
Statmt-news_commentary-18.1	244,831	244,831
OPUS-neulab_tedtalks-v1	96,738	96,738
OPUS-ted2020-v1	153,227	153,227
OPUS-opensubtitles-v2024	36,408,370	168,401
OPUS-dgt-v4	3,048,670	168,401
OPUS-europarl-v8	568,589	168,402
Total		1,000,000
ENG-ARA		
OPUS-globalvoices-v2018q4	59,196	59,196
Statmt-news_commentary-18.1	193,671	193,671
Statmt-tedtalks-2_clean	341,887	149,426
OPUS-ted2020-v1	403,716	149,426
OPUS-qed-v2.0a	500,898	149,426
OPUS-opensubtitles-v2024	87,893,568	149,426
OPUS-multiun-v1	9,759,125	149,429
Total		1,000,000
JPN-ZHO		
Statmt-news_commentary-18.1	1,625	1,625
OPUS-ted2020-v1	15,982	15,982
Neulab-tedtalks_train-1	5,159	5,159
KECL-paracrawl-2wmt24	4,602,328	488,617
OPUS-opensubtitles-v2024	1,267,153	488,617
Total		1,000,000

Table 1: Dataset statistics for WMT 2025 Model Compression shared task training data. Total Size indicates the original dataset sizes, while Samples indicates the actual number of translation pairs used post-training.

Language Pair	Dataset	Samples
CES-DEU	newstests2019 (WMT 2024)	1,997
ENG-ARA	wmttest2024 (WMT 2024)	721
JPN-ZHO	WMT24++ (Deutsch et al., 2025)	998

Table 2: Test set statistics in terms of number of sentence pairs.

4 Experimental Setup

4.1 Models

Following the requirements of the constrained track of the shared task, we used the Aya Expanse 8B model as our foundation. This instruction-tuned model served both as our starting point for compression and as the primary baseline for performance comparison. We preserved the capabilities of the original model by not applying any additional training or modification to the base model prior to compression.

4.2 Corpora

Following the constrained track requirements, we sourced all training data from the WMT 2025 MT task data releases[1]. Our data selection strategy leveraged the available parallel corpora for each language pair, sampling from diverse sources of varying quality and domains. We arbitrarily selected one million translation instruction pairs per language combination as a compromise between coverage and reducing post-training computational time.

We provide a detailed breakdown of the original and sampled datasets in Table 1. Our training data consisted of one million translation instruction pairs for each of the three language pairs (Czech-German, English-Arabic, and Japanese-Chinese), yielding a total of 3 million translation instructions. We detail the specific instruction template used for translation post-training in Appendix B.

For evaluation, we selected test sets based on data released for WMT 2024[2]. Table 2 reports the number of translation pairs for each language pair and test set.

4.3 Compression

For the evolutionary search process, we used 16 randomly selected sentences from the monolingual portion of ParaCrawl for each target language, resulting in a total of 96 sentences as calibration data. We executed GeLaCo with the same configuration parameters as defined in the original work, running for 10,000 evolutionary steps with a single compression objective for each target ratio.

Using GeLaCo, we compressed the original 32-layer, 8-billion parameter model at three levels: 0.25 compression yielded 24 layers and approximately 6.2 billion parameters; 0.50 compression resulted in 16 layers and 4.5 billion parameters; and 0.75 compression produced 8 layers and 2.8 billion parameters.

For quantization, we employed the bitsandbytes library[3] to generate 8-bit and 4-bit with double quantization variants, providing additional compression beyond the structural layer reduction.

4.4 Post-training

We leveraged the 3 million translation instructions to perform both Supervised Fine-Tuning and Generalized Knowledge Distillation on the compressed models. For computational efficiency, we conducted all training using DeepSpeed with ZeRO Stage 3 optimization (Rajbhandari et al., 2020).

Due to the substantial computational requirements of GKD, we applied this technique exclusively to our smallest compressed model (0.75 compression ratio), while SFT was performed across all compression levels. The detailed hyperparameters for both SFT and GKD training, as well as the DeepSpeed ZeRO configuration, are provided in Appendix C.

4.5 Inference

We used vLLM (Kwon et al., 2023) for efficient inference across all experiments. For few-shot learning, we used 5 examples per evaluation. In the static few-shot setup, we randomly selected 5 translation instructions for each language pair from the training set. For dynamic few-shot learning, we performed BM25 retrieval over a subset of 10,000 training instances per language, selecting the 5 most similar translations to each source sentence. For retrieval, we used the Okapi BM25 implementation from Rank-BM25[4], configured with a minimum token length of 4 characters and whitespace tokenization.

[1] https://www2.statmt.org/wmt25/mtdata/
[2] https://data.statmt.org/wmt24/
[3] https://github.com/bitsandbytes-foundation/bitsandbytes
[4] https://github.com/dorianbrown/rank_bm25

Method	Size (GiB)	Inference	Time (s)	CES-DEU		ENG-ARA		JPN-ZHO	
				chrF	COMET	chrF	COMET	chrF	COMET
aya-expanse-8b	14.96	Zero-shot	88.67	54.1	0.8476	39.6	0.7699	23.5	0.8142
		Few-shot	60.32	53.7	0.8458	39.2	0.7709	22.5	0.8140
		RAG	619.14	53.5	0.8461	39.5	0.7721	23.0	0.8117
GeLaCo 0.25 - SFT	11.71	Zero-shot	78.36	51.2	0.8304	33.7	0.7327	17.8	0.7611
		Few-shot	83.24	51.2	0.8293	34.5	0.7300	18.0	0.7622
		RAG	616.42	51.3	0.8289	33.8	0.7200	17.7	0.7612
GeLaCo 0.50 - SFT	8.46	Zero-shot	73.61	48.3	0.7964	30.4	0.7105	13.0	0.7039
		Few-shot	66.74	48.1	0.7964	29.5	0.6976	12.3	0.69597
		RAG	641.88	48.2	0.7967	29.6	0.6945	12.0	0.6942
GeLaCo 0.75 - SFT	5.21	Zero-shot	62.98	40.7	0.6578	19.5	0.588	5.1	0.5499
		Few-shot	62.97	42.0	0.6612	18.9	0.5788	5.5	0.556
		RAG	588.99	40.2	0.6563	19.1	0.5797	5.2	0.5556
GeLaCo 0.75 - GKD	5.21	Zero-shot	49.79	50.3	0.7799	32.0	0.6964	16.7	0.7178
		Few-shot	53.14	14.5	0.3662	9.30	0.4271	1.3	0.4006
		RAG	440.86	13.4	0.3930	5.9	0.4377	1.2	0.3911

Table 3: Translation performance comparison across compression ratios. Results show chrF and COMET scores for zero-shot, few-shot, and RAG inference approaches across all three language pairs. SFT indicates fine-tuning and GKD indicates Generalized Knowledge Distillation post-training.

4.6 Evaluation

We evaluated translation quality using chrF (Popović, 2015) and COMET (Rei et al., 2020) with the Unbabel/wmt22-comet-da model[5]. For model size measurements, we report the VRAM usage during vLLM inference.

For inference speed evaluation, we report timing measurements computed using a batch size of 4,096, using bfloat16 precision for the non-quantized models. To ensure consistent timing comparisons, we excluded model loading times from our reported measurements due to the high variability and inconsistency that we observed during this phase and conducted 5 runs for each measurement to reduce variability and report the average results.

4.7 Hardware

For the GeLaCo compression process and inference experiments, we used a single NVIDIA L40 GPU with 48GB of vRAM. For post-training we used 4 NVIDIA H100 GPUs with 80GB of vRAM each for both SFT and GKD.

5 Results

We present our experimental results analyzing the impact of compression ratios, post-training approaches, and quantization on translation quality, model size, and inference speed.

5.1 Compression and Post-training

Table 3 presents the results for our baseline Aya Expanse 8B model and the compressed variants at compression ratios of 0.25, 0.50, and 0.75, along with their corresponding inference times for processing all three test sets. The results demonstrate the expected trade-off between model compression and translation quality across all evaluation settings.

As expected, model performance degrades progressively with increased compression levels, even after fine-tuning recovery. This pattern of declining performance with higher compression ratios is consistent across all language pairs and evaluation metrics. A notable result is that GKD demonstrates superior performance recovery compared to fine-tuning. For zero-shot translation, GKD at 0.75 compression shows improvements of +2.0, +1.6, and +3.7 chrF points for CES-DEU, ENG-ARA, and JPN-ZHO respectively compared to the 0.50 SFT model, while also showing an improvement of +1.4 COMET points in JPN-ZHO.

In-context Learning. Regarding in-context learning strategies, both static few-shot sampling and dynamic similarity-based retrieval yield results comparable to the zero-shot approach for the SFT models. However, the GKD-trained model presents a different behaviour, where ICL

[5]https://huggingface.co/Unbabel/wmt22-comet-da

Method	Quant.	Size (GiB)	CES-DEU		ENG-ARA		JPN-ZHO	
			chrF	COMET	chrF	COMET	chrF	COMET
	-	14.96	54.1	0.8476	39.6	0.7699	23.5	0.8142
aya-expanse-8b	Q8	8.46	54.2	0.8479	39.5	0.7661	23.5	0.8111
	Q4	5.31	53.9	0.8452	38.9	0.7661	22.8	0.8116
	-	11.71	51.2	0.8304	33.7	0.7327	17.8	0.7611
GeLaCo 0.25 - SFT	Q8	6.83	51.5	0.8318	33.7	0.7324	17.8	0.7606
	Q4	4.47	50.3	0.8242	32.1	0.7193	16.4	0.7477
	-	8.46	48.3	0.7964	30.4	0.7105	13.0	0.7039
GeLaCo 0.50 - SFT	Q8	5.21	48.2	0.7951	30.8	0.7125	12.7	0.6980
	Q4	3.63	46.5	0.7813	27.7	0.6878	12.3	0.6886
	-	5.21	40.7	0.6578	19.5	0.588	5.1	0.5499
GeLaCo 0.75 - SFT	Q8	3.58	40.3	0.6514	19.0	0.5782	4.9	0.5476
	Q4	2.79	35.3	0.6066	16.9	0.5491	4.5	0.5276
	-	5.21	50.3	0.7799	32.0	0.6964	16.7	0.7178
GeLaCo 0.75 - GKD	Q8	3.58	49.8	0.7789	32.5	0.6968	16.6	0.7204
	Q4	2.79	49.7	0.7756	31.1	0.6884	16.6	0.7174

Table 4: Impact of quantization on compressed model performance. Results compare 8-bit (Q8) and 4-bit (Q4) quantization using zero-shot translation across all language pairs.

methods fail dramatically. The GKD model's few-shot performance drops drastically across all language pairs, from 50.3 to 14.5 chrF for CES-DEU, 32.0 to 9.3 for ENG-ARA, and 16.7 to 1.3 for JPN-ZHO. Future work should address the drastic loss of in-context learning capabilities in GKD-trained models.

While we observe the expected reduction in processing time with model compression, from 88.67 seconds for the baseline to 49.79 seconds for the 0.75 GKD model in zero-shot setting, timing measurements showed unexpected variations where few-shot inference was occasionally faster than zero-shot despite the longer context, reflecting the inherent difficulties in timing measurements. Nevertheless, the computational overhead of the RAG approach consistently requires an order of magnitude more time, primarily due to the retrieval process overhead rather than the translation itself.

5.2 Quantization Results

Table 4 presents the results of applying quantization techniques to the GeLaCo compressed models, examining the effects of 8-bit and 4-bit quantization on model size and translation performance in a zero-shot setting.[6]

Across all GeLaCo variants, 8-bit quantization (Q8) reduces model sizes by approximately 42% while maintaining stable translation performance. For the 0.25 compressed model, Q8 reduces the size from 11.71 GiB to 6.83 GiB with minimal quality impact. The 0.50 compressed model follows a similar pattern, achieving a size reduction from 8.46 GiB to 5.21 GiB with marginal quality variations across language pairs. The 0.75 models also benefit from Q8 quantization, with both SFT and GKD variants reducing from 5.21 GiB to 3.58 GiB while preserving competitive performance.

4-bit quantization (Q4) enables more aggressive compression but introduces more noticeable quality degradation. For the 0.25 compressed model, Q4 reduces the size to 4.47 GiB while incurring chrF drops of 0.9, 1.6, and 1.4 points for CES-DEU, ENG-ARA, and JPN-ZHO respectively. This pattern intensifies with higher compression ratios, where the 0.75 SFT model with Q4 shows significant performance drops, particularly evident in the chrF scores falling to 35.3, 16.9, and 4.5.

A notable result emerges with the GKD variant, which demonstrates superior robustness to quantization. The 0.75 GKD model maintains competitive performance even with aggressive Q4 quantization, achieving chrF scores of 49.7, 31.1, and 16.6, substantially outperforming the corresponding SFT variant under the same quantization settings. The combination of layer compression and quantization enables the creation of extremely compact models, with the 0.75 GKD model reach-

[6]Complete quantization results including few-shot and RAG are provided in Appendix A

ing 2.79 GiB with Q4, representing an 81% reduction from the original baseline while retaining reasonable translation capabilities.

Given that quantization produces only marginal quality degradation while achieving substantial size reductions across all compression levels, we selected each of the Q8 and Q4 variants as our final submissions to the shared task. Specifically, we submitted the 0.25, 0.50, 0.75 SFT models and the 0.75 GKD model unquantized, Q8 and Q4 variants, with the 0.75 GKD Q4 model being designated as our primary submission due to its optimal balance of compression efficiency and translation quality preservation.

6 Conclusions

This work presented our approach to the WMT 2025 Model Compression shared task, focusing on compressing the Aya Expanse 8B model for machine translation across Czech-German, English-Arabic, and Japanese-Chinese language pairs within the constrained setting of the task. We employed GeLaCo, an evolutionary algorithm for layer collapse operations, combined with post-training techniques and quantization, to achieve substantial model size reduction while maintaining competitive translation performance.

Our experimental results demonstrated that compressed models can be successfully recovered through targeted post-training techniques. Generalized Knowledge Distillation consistently outperformed traditional fine-tuning for performance recovery across all three language pairs at the 0.75 compression ratio where it was applied. The combination of layer compression with 4-bit quantization achieved an 81% reduction in model size (from 14.96 GiB to 2.79 GiB) while preserving reasonable translation quality, making such models viable for resource-constrained scenarios.

Acknowledgments

This work was partially supported by the Department of Economic Development and Competitiveness of the Basque Government (SPRI Group) through funding for project IKUN (KK2024/00064).

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.

Duarte Miguel Alves, José Pombal, Nuno M Guerreiro, Pedro Henrique Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and Andre Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. In *First Conference on Language Modeling*.

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. SliceGPT: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*.

Tom Brown et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Yupeng Chang et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. 2025. Streamlining redundant layers to compress large language models. In *The Thirteenth International Conference on Learning Representations*.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, Acyr Locatelli, Hangyu Lin, Dwarak Talupuru, Bharat Venkitesh, David Cairuz, Bowen Yang, Tim Chung, Wei-Yin Ko, Sylvie Shang Shi, Amir Shukayev, Sammie Bae, Aleksandra Piktus, Roman Castagné, Felipe Cruz-Salinas, Eddie Kim, Lucas Crawhall-Stein, Adrien Morisot, Sudip Roy, Phil Blunsom, Ivan Zhang, Aidan Gomez, Nick Frosst, Marzieh Fadaee, Beyza Ermis, Ahmet Üstün, and Sara Hooker.

2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier.

Daniel Deutsch, Eleftheria Briakou, Isaac Rayburn Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus Freitag. 2025. WMT24++: Expanding the language coverage of WMT24 to 55 languages & dialects. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12257–12284, Vienna, Austria. Association for Computational Linguistics.

Marco Gaido, Thamme Gowda, Roman Grundkiewicz, and Matteo Negri. 2025. Findings of the WMT25 Model Compression Shared Task: Early Insights on Compressing LLMs for Machine Translation. In *Proceedings of the Tenth Conference on Machine Translation*, Suzhou, China.

R.M. Gray and D.L. Neuhoff. 1998. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinthór Steingrímsson, and Vilém Zouhar. 2024. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Masaaki Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, Mariya Shmatova, and Jun Suzuki. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-pruner: On the structural pruning of large language models. *Advances in Neural Information Processing Systems*, 36:21702–21720.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M Guerreiro, Ricardo Rei, Duarte M Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, et al. 2025. Eurollm: Multilingual language models for europe. *Procedia Computer Science*, 255:53–62.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and weipeng chen. 2025. ShortGPT: Layers in large language models are more redundant than you expect.

David Ponce, Thierry Etchegoyhen, and Javier Del Ser. 2025. Gelaco: An evolutionary approach to layer compression. *arXiv preprint arXiv:2507.10059*.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference* for High Performance Computing, Networking, Storage and Analysis, pages 1–16. IEEE.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Oliver Sieberling, Denis Kuznedelev, Eldar Kurtic, and Dan Alistarh. 2024. EvoPress: Towards optimal dynamic model compression via evolutionary search. *arXiv preprint arXiv:2410.14649*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Shengkun Tang, Oliver Sieberling, Eldar Kurtic, Zhiqiang Shen, and Dan Alistarh. 2025. Darwinlm: Evolutionary structured pruning of large language models. *arXiv preprint arXiv:2502.07780*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*.

Yifei Yang, Zouying Cao, and Hai Zhao. 2024. LaCo: Large language model pruning via layer collapse. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6401–6417, Miami, Florida, USA. Association for Computational Linguistics.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024a. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024b. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

A Full Results

Quant.	Method	Size (GiB)	Inference	CES-DEU		ENG-ARA		JPN-ZHO	
				chrF	COMET	chrF	COMET	chrF	COMET
-	aya-expanse-8b	14.96	Zero-shot	54.1	0.8476	39.6	0.7699	23.5	0.8142
			Few-shot	53.7	0.8458	39.2	0.7709	22.5	0.8140
			RAG	53.5	0.8461	39.5	0.7721	23.0	0.8117
	GeLaCo 0.25 - SFT	11.71	Zero-shot	51.2	0.8304	33.7	0.7327	17.8	0.7611
			Few-shot	51.2	0.8293	34.5	0.7300	18.0	0.7622
			RAG	51.3	0.8289	33.8	0.7200	17.7	0.7612
	GeLaCo 0.50 - SFT	8.46	Zero-shot	48.3	0.7964	30.4	0.7105	13.0	0.7039
			Few-shot	48.1	0.7964	29.5	0.6976	12.3	0.69597
			RAG	48.2	0.7967	29.6	0.6945	12.0	0.6942
	GeLaCo 0.75 - SFT	5.21	Zero-shot	40.7	0.6578	19.5	0.588	5.1	0.5499
			Few-shot	42.0	0.6612	18.9	0.5788	5.5	0.556
			RAG	40.2	0.6563	19.1	0.5797	5.2	0.5556
	GeLaCo 0.75 - GKD	5.21	Zero-shot	50.3	0.7799	32.0	0.6964	16.7	0.7178
			Few-shot	14.5	0.3662	9.30	0.4271	1.3	0.4006
			RAG	13.4	0.3930	5.9	0.4377	1.2	0.3911
Q8	aya-expanse-8b	14.96	Zero-shot	54.2	0.8479	39.5	0.7661	23.5	0.8111
			Few-shot	53.9	0.8472	39.3	0.7682	22.5	0.8153
			RAG	53.7	0.8466	39.4	0.7713	22.9	0.8113
	GeLaCo 0.25 - SFT	11.71	Zero-shot	51.5	0.8318	33.7	0.7324	17.8	0.7606
			Few-shot	51.5	0.8316	34.6	0.7304	18.1	0.7641
			RAG	51.4	0.8299	33.2	0.7205	18.0	0.7616
	GeLaCo 0.50 - SFT	8.46	Zero-shot	48.2	0.7951	30.8	0.7125	12.7	0.6980
			Few-shot	48.6	0.7953	29.7	0.6983	12.4	0.6925
			RAG	48.9	0.7976	29.8	0.6928	12.2	0.6968
	GeLaCo 0.75 - SFT	5.21	Zero-shot	40.3	0.6514	19.0	0.5782	4.9	0.5476
			Few-shot	39.8	0.6516	17.6	0.5678	5.4	0.5546
			RAG	40.3	0.6537	17.3	0.5692	5.0	0.5472
	GeLaCo 0.75 - GKD	5.21	Zero-shot	49.8	0.7789	32.5	0.6968	16.6	0.7204
			Few-shot	14.2	0.3732	10.1	0.4374	1.4	0.4049
			RAG	13.0	0.3923	6.1	0.4378	1.3	0.3948
Q4	aya-expanse-8b	14.96	Zero-shot	53.9	0.8452	38.9	0.7661	22.8	0.8116
			Few-shot	53.2	0.8431	39.1	0.7681	21.7	0.8140
			RAG	53.2	0.8425	39.1	0.7685	22.4	0.8076
	GeLaCo 0.25 - SFT	11.71	Zero-shot	50.3	0.8242	32.1	0.7193	16.4	0.7477
			Few-shot	50.3	0.8239	32.7	0.7155	16.7	0.7477
			RAG	50.4	0.8238	31.6	0.7076	16.6	0.7496
	GeLaCo 0.50 - SFT	8.46	Zero-shot	46.5	0.7813	27.7	0.6878	12.3	0.6886
			Few-shot	47.3	0.7807	28.8	0.6801	12.4	0.6845
			RAG	47.4	0.7831	27.8	0.6733	12.5	0.6861
	GeLaCo 0.75 - SFT	5.21	Zero-shot	35.3	0.6066	16.9	0.5491	4.5	0.5276
			Few-shot	37.0	0.6164	16.9	0.5571	4.5	0.5311
			RAG	37.3	0.6196	16.6	0.552	4.5	0.5353
	GeLaCo 0.75 - GKD	5.21	Zero-shot	49.7	0.7756	31.1	0.6884	16.6	0.7174
			Few-shot	12.9	0.3767	9.9	0.3931	1.4	0.4009
			RAG	12.3	0.3874	5.9	0.4277	1.4	0.3949

Table 5: Complete experimental results across all models, quantization settings, and inference approaches.

B Translation Instruction Template

The following template illustrates the format used for translation instructions. At inference time, only the user message is provided to the model. The instruction prompt and language names are always specified in English. In the template, SOURCE_LANGUAGE and TARGET_LANGUAGE represent the English names of the source and target languages (e.g., "Czech", "German", "English", "Arabic", "Japanese", or "Chinese"), INPUT_SENTENCE contains the text to be translated in the source language, and TARGET_SENTENCE contains the corresponding translation in the target language.

Instruction Template

```
"messages": [
    {
        "role": "user",
        "content": "Translate from SOURCE_LANGUAGE to TARGET_LANGUAGE:\nINPUT_SENTENCE"
    },
    {
        "role": "assistant",
        "content": "TARGET_SENTENCE"
    }
]
```

C Training Hyperparameters

For both supervised fine-tuning and generalized knowledge distillation, we employed the SFTTrainer and GKDTrainer implementations from the TRL[7] library. All training was conducted using DeepSpeed with ZeRO Stage 3 optimization for efficient memory management across multiple GPUs. The specific hyperparameters used for each training approach are detailed below.

SFT Training Hyperparameters

```
--learning_rate 2.0e-5
--num_train_epochs 3
--packing
--per_device_train_batch_size 8
--gradient_accumulation_steps 4
--gradient_checkpointing
--bf16 True
```

GKD Training Hyperparameters

```
--learning_rate 2.0e-5
--per_device_train_batch_size 4
--gradient_accumulation_steps 8
--bf16 True
--logging_steps 25
```

DeepSpeed ZeRO Configuration

```
compute_environment: LOCAL_MACHINE
debug: false
deepspeed_config:
  deepspeed_multinode_launcher: standard
  offload_optimizer_device: none
  offload_param_device: none
  zero3_init_flag: true
  zero3_save_16bit_model: true
  zero_stage: 3
distributed_type: DEEPSPEED
downcast_bf16: 'no'
machine_rank: 0
main_training_function: main
mixed_precision: bf16
num_machines: 1
num_processes: 8
rdzv_backend: static
same_network: true
tpu_env: []
tpu_use_cluster: false
tpu_use_sudo: false
use_cpu: false
```

[7]https://huggingface.co/docs/trl/index