Findings of the WMT 2025 Shared Task on Model Compression: Early Insights on Compressing LLMs for Machine Translation

Marco Gaido FBK
Thamme Gowda Microsoft
Roman Grundkiewicz Microsoft
Matteo Negri FBK

{mgaido,negri}@fbk.eu, {thammegowda,rogrundk}@microsoft.com

Abstract

We present the results of the first edition of the Model Compression shared task, organized as part of the 10th Conference on Machine Translation (WMT25). The task challenged participants to compress Large Language Models (LLMs) toward enabling practical deployment in resource-constrained scenarios, while minimizing loss in translation performance. In this edition, participants could choose to compete in either a constrained track, which required compressing a specific model (Aya Expanse 8B) evaluated on a limited set of language pairs (Czech→German, Japanese→Chinese, and English→Arabic), or an unconstrained track, which placed no restrictions on the model and allowed submissions for any of the 15 language directions covered by the General MT task. We received 12 submissions from 3 teams, all in the constrained track. They proposed different compression solutions and covered various language combinations. Evaluation was conducted separately for each language, measuring translation quality using COMET and MetricX, model size, and inference speed on an Nvidia A100 GPU.

1 Introduction

Large Language Models (LLMs) have demonstrated exceptional performance across a wide range of tasks. However, efforts to enhance their capabilities, by expanding language coverage, integrating multimodal data, and improving task generalization, have led to a dramatic increase in both model size and computational demands (Zhu et al., 2024). This rapid growth poses significant challenges for real-world deployment, particularly in resource-constrained environments such as mobile devices, embedded systems, and edge computing platforms, where low-latency, on-device processing is often required. Compressing foundation models is therefore more than a technical pursuit: it is a strategic priority with implications for global accessibility[1] and the sustainability of computational and environmental costs. Striking the right balance between performance, compactness, and efficiency is thus essential to make LLMs truly ubiquitous and beneficial for everyone, regardless of location or access to high-end infrastructure.

It is with this long-term goal in mind that, following the analogous task for speech translation in the IWSLT 2025 campaign (Abdulmumin et al., 2025), the new Model Compression shared task was introduced at WMT 2025.[2] This initiative follows three editions of the shared task at the Workshop on Machine Translation and Generation (Birch et al., 2018; Hayashi et al., 2019; Heafield et al., 2020) and two editions of the shared task on Efficient Translation (Heafield et al., 2021, 2022). It revives earlier focus on the efficiency of machine translation, while updating it to reflect the current AI landscape with the rise of general-purpose LLMs.

In this context, our aim is to provide a timely evaluation of compression techniques for general-purpose LLMs within the specific task of machine translation. This setting offers a valuable opportunity to explore key research questions, such as:

- *To what extent can the over-parameterization of LLMs—originally pursued to enable generalization, robustness, task flexibility, and broad language coverage—be reduced in favor of compactness and efficiency, while preserving MT quality?*
- *How do different compression techniques, with varying degrees of aggressiveness, impact translation quality in such settings?*

[1] In the U.S. around 15% of adults rely exclusively on mobile devices to access the internet (https://www.pewresearch.org/internet/fact-sheet/mobile/), and it is even more pronounced in developing regions (https://www.eib.org/en/essays/african-digital-infrastructure).

[2] https://www2.statmt.org/wmt25/

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 676–686
November 8–9, 2025. ©2025 Association for Computational Linguistics

2 Task Description

The goal of the Model Compression task is to reduce the size of a general-purpose LLM while preserving a strong balance between compactness and MT performance. This section provides a brief overview of how the first round of the task was structured, focusing on the proposed tracks, data conditions, and evaluation methodology.

2.1 Tracks

Participants could choose between two tracks: constrained and unconstrained.

The **constrained** track was designed to ensure a level playing field by establishing uniform conditions across all participants, allowing for directly comparable results. It focused on the compression of a specific model in a fixed language setting. The model selected for this purpose was Aya Expanse 8B,[3] chosen for its permissive license (CC-BY-NC 4.0) and its favorable trade-off between the size (8 billion parameters; approximately 16 GB in FP16 precision) and performance.

In the constrained settings, we measured performance across three language pairs: Czech→German, Japanese→Chinese, English→Arabic. These pairs were selected to provide a sufficiently diverse coverage of language families and scripts. Submissions were allowed for any of these directions. Any model compression technique e.g., pruning (Frankle and Carbin, 2019; Frankle et al., 2020), quantization (Devlin, 2017), or distillation (Kim and Rush, 2016), was permitted, provided that the final compressed model remained closely derived from Aya Expanse 8B. For instance, in the case of distillation, student models had to be obtained through compression of Aya Expanse 8B (e.g., by pruning or quantizing it) to qualify for the constrained track. Otherwise, we would consider such systems as unconstrained.

The **unconstrained** track provided participants with complete freedom to compress any model of their choice and apply it to any of the 15 language directions covered by the WMT25 General MT task (GenMT) (Kocmi et al., 2025). As in the constrained track, separate rankings were planned for each language direction.

[3] https://huggingface.co/CohereLabs/aya-expanse-8b

2.2 Data

Data usage policies were aligned with those of the GenMT task. Participants were therefore allowed to calibrate and fine-tune their compressed models using the publicly available datasets released for this year's round,[4] as well as test sets from previous WMT editions.

2.3 Evaluation

Submissions were evaluated[5] along three key dimensions:

- **Translation quality** measured using the same automatic metrics employed in the GenMT task;
- **Model size** as disk space footprint;
- **Inference speed** as the average number of output tokens produced per second when processing the test set.

All three were considered both *independently* and *jointly*. We report Pareto frontier rankings to visualize system differences through quality–size, quality–speed and size–speed plots. Since we received multiple submissions only for Czech→German, this type of visualization was only feasible for that language direction.

To ensure a fair and informative evaluation, we create a homogeneous hardware environment for running the submitted systems. We used machines with a single Nvidia A100 GPU having 80GB of VRAM, AMD EPYC CPU with 96 cores, and 866GB RAM.

2.4 Submission

Participating teams were asked to provide a link to a Docker image containing all necessary software and model files for translation, along with basic information about the maximum batch size supported by their model(s) under the specified hardware configuration. Upon request, we also offered storage space to teams who needed it or preferred to upload their models externally to their institutional infrastructure.

3 Participants

Three teams submitted systems to the task, as summarized in Table 1. The organizers also included baseline systems. Below, we provide a brief

[4] https://www2.statmt.org/wmt25/mtdata/

[5] Scripts used for evaluation are available at: https://github.com/thammegowda/wmt25-model-compression

Institution	Submission	Track	No. Sub.	Languages
Stevens Institute of Technology, Rice University, Lambda Inc.	AyaQ	Constr.	1	cs-de
Stevens Institute of Technology, Rice University, Lambda Inc.	LeanAya	Constr.	1	cs-de
Trinity College Dublin (Ponce et al., 2025b)	TCD-Kreasof	Constr.	3	cs-de
Vicomtech (Ponce et al., 2025b)	Vicomtech	Constr.	7	cs-de, jp-zh, en-ar
Organizers (compressed baseline model)	BitsAndBytes	Constr.	4	*as baseline*

Table 1: Participants in the WMT 2025 Model Compression shared task with the number of submitted system variants and declared language support.

Submission	Description
base	Base Aya Expanse 8B in 16bit
bnb-8bit	8bit integer
bnb-4bit-fp4	4bit FP4
bnb-4bit-nf4	4bit NF4
bnb-4bit-nf4-2q	4bit NF4, double-quantization

Table 2: Baseline and BitsAndBytes (Dettmers et al., 2022, 2023) systems submitted by the organizers.

overview of the proposed approaches, all developed within the constrained track.

AyaQ[6] This participation employs GPTQ 4-bit quantization (Frantar et al., 2023) with a group size of 32 to enable efficient and scalable LLM inference. The WMT dataset is used as calibration data to guide the quantization process, ensuring the compressed model retains high accuracy on language understanding and generation tasks. The quantized models are integrated through the LLM Compressor framework (AI and vLLM Project, 2024), which streamlines conversion and metadata management. The setup is fully compatible with vLLM (Kwon et al., 2023), a high-throughput inference engine optimized for GPU deployment, enabling fast and memory-efficient execution with minimal performance loss. This approach demonstrates how structured quantization, targeted calibration, and system-level integration can enable practical, production-ready LLM deployment.

LeanAya This participation is based on LeanQuant (Loss-Error-Aware Network Quantization, (Zhang and Shrivastava, 2025)), an accurate, versatile, and scalable quantization method. Existing iterative loss-error-based quantization techniques typically rely on min-max affine grids, which often degrade model quality due to outliers in the inverse Hessian diagonals. LeanQuant overcomes this limitation by learning loss-error-aware quantization grids instead of using fixed, non-adaptive ones. This approach not only improves accuracy but also supports a wider range of quantization schemes, including both affine and non-uniform, enhancing compatibility across diverse deployment frameworks.

TCD-Kreasof (Moslem et al., 2025) This participation employs iterative layer pruning to incrementally identify and remove layers that contribute least to translation quality, one at a time. Layer importance is assessed by measuring translation performance with each layer individually removed. After pruning the least critical layer, the evaluation is repeated on the remaining ones until the target pruning level is reached. The resulting pruned model was then fine-tuned on 100k sentences from the News Commentary dataset. This process produced three submissions: the primary one is a 24-layer model with 6.28B parameters, while the two contrastive submissions are 20-layer and 16-layer models, with 5.41B and 4.54B parameters, respectively.

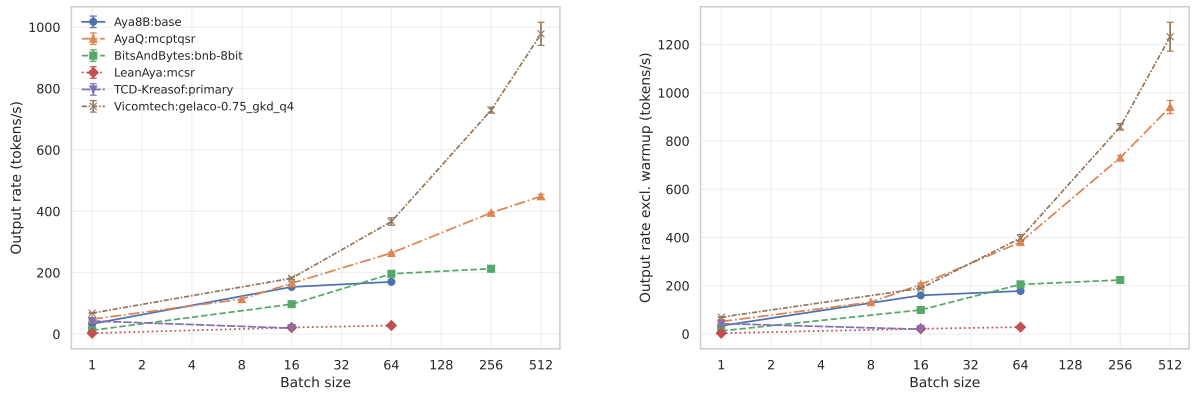
Vicomtech (Ponce et al., 2025b) This participation employs GeLaCo (Ponce et al., 2025a), an evolutionary approach to LLM compression based on layer merging operations. Models are compressed at three ratios (0.25, 0.50, and 0.75), representing the proportion of original layers collapsed through differential weight merging. To recover performance after compression, over 3 million translation instructions (1 million per language) from a subset of WMT25 translation data are used. For the 0.25 and 0.50 compression levels, models are fine-tuned on this data, while the 0.75 model is trained using General Knowledge Distillation (GKD (Tan et al., 2023)). Additionally, post-training quantization is applied using the `bitsandbytes` library[7] to further reduce model size to 8-bit and 4-bit precision. The primary submission (`gelaco-0.75_gkd_q4`)

[6] We did not receive system description papers for AyaQ and LeanAya submissions.

[7] https://github.com/bitsandbytes-foundation/bitsandbytes

		English→Arabic		Japanese→Chinese		Czech→German		
Submission	**System**	**COMET↑**	**MetricX↓**	**COMET↑**	**MetricX↓**	**COMET↑**	**MetricX↓**	**#Halluc.**
Baseline	base	25.4	8.20	44.5	6.44	55.3	5.08	83
BitsAndBytes	**bnb-8bit**	25.2	8.33	44.4	6.49	55.6	5.08	86
	bnb-4bit-fp4	24.4	8.26	43.6	6.54	54.5	5.24	153
	bnb-4bit-nf4	25.4	8.25	44.6	6.43	55.7	5.15	103
	bnb-4bit-nf4-2q	25.4	8.25	44.6	6.41	55.5	5.15	106
AyaQ	**mcptqsr**	–	–	–	–	40.3	8.70	200
LeanAya	**mcsr**	–	–	–	–	53.2	5.36	66
TCD-Kreasof	**primary**	–	–	–	–	39.9	7.93	78
	contrastive1	–	–	–	–	32.4	9.49	102
	contrastive2	–	–	–	–	21.4	14.53	335
Vicomtech	gelaco-0.25_ft_q4	20.9	9.80	38.7	8.55	41.2	7.52	37
	gelaco-0.25_ft_q8	22.0	9.27	39.0	8.42	44.4	6.75	42
	gelaco-0.50_ft_q4	18.0	12.10	31.4	10.12	31.0	9.82	94
	gelaco-0.50_ft_q8	17.9	11.57	32.2	10.15	33.7	9.24	52
	gelaco-0.75_gkd	16.1	13.90	31.8	9.93	30.6	11.03	198
	gelaco-0.75_gkd_q4	16.7	13.98	32.2	9.86	31.1	11.04	187
	gelaco-0.75_gkd_q8	15.7	13.57	31.5	9.87	31.1	10.82	197

Table 3: Translation quality metric scores on the official WMT25 GenMT test sets. XCOMET-XL and MetricX-24-Hybrid-XL scores. AyaQ and LeanAya declared support only for Czech→German. TCD-Kreasof systems did not allow to generate outputs for other languages.



(a) Total output token rates.

(b) Output rates excluding warmup times.

Figure 1: Inference speed as tokens/s when translating the entire Czech→GermanWMT25 test set. Mean and standard deviation across 3 runs, reported for various batch sizes. Primary submissions only for readability.

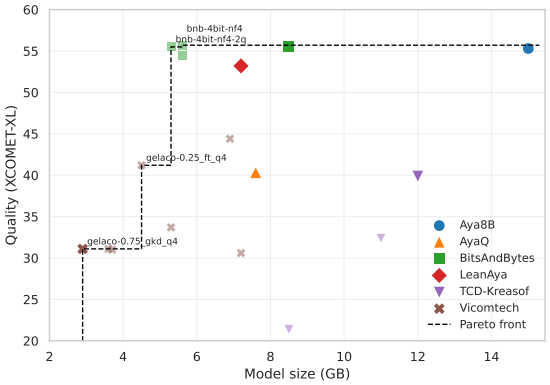
combines evolutionary layer collapse, knowledge distillation, and quantization to achieve substantial size reduction while maintaining reasonable translation performance.

Baselines As a reference system (see Table 2), we included the unmodified Aya Expanse 8B model (FP16, 16.1GB) (Dang et al., 2024) and a family of runtime–quantized variants created using the Hugging Face integration of the `bitsandbytes` library (Dettmers et al., 2022, 2023), without the use of vLLM. The baselines were not fine-tuned or adapted on the task data; their purpose was to anchor the quality–size–speed trade-offs for submitted systems. Quantized versions include 8-bit and 4-bit modes, with two 4-bit quantization data types: NF4 (normal floating point) and FP4.

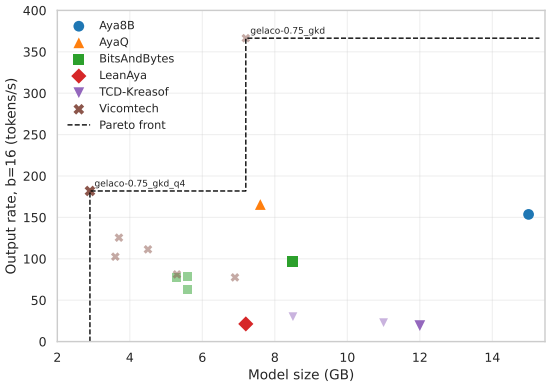
The organizers' baselines illustrate the performance one can obtain from (i) the full reference model, and (ii) straightforward, widely available post-training quantization strategies, against which more sophisticated compression pipelines can be directly compared in terms of translation quality, memory footprint, and decoding speed.

4 Results

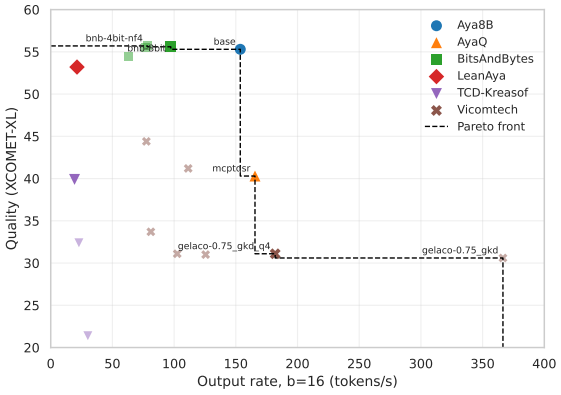
We evaluate systems' performance on the official WMT25 GenMT test sets, which comprise between 332 and 456 paragraphs for each of the three considered language pairs. Because we received submissions to the constrained track only and most



(a) Model size and COMET scores.



(b) Model size and total output token rate.



(c) Total output rate and COMET scores.

Figure 2: Comparison of translation quality, model size and inference speed for batch size 16. The staircase shows the Pareto frontline.

submissions focused on Czech→German, we primarily report results for this language pair unless mentioned otherwise.

We noticed that most of the systems may suffer from hallucinated content, which affects both translation quality and decoding speed, and complicates system comparisons. To minimize the impact of hallucinations, we segment the original paragraph-level test data at newline characters and remove empty lines. For Czech-German, this results in a test set with 2,868 segments.

To further investigate the potential impact of hallucinations, we also evaluate on a subset of the test set that only includes 1,928 segments where none of the systems exhibits hallucinations under any batch setting. This subset makes it possible to assess system performance in terms of inference speed more accurately, by excluding the distortions that hallucinations would introduce into cross-system comparisons. For brevity, analysis on the hallucination-free test data is presented in Appendix A.

Below, we briefly discuss the results in terms of quality, model size and speed. The detailed results across more benchmark settings are provided in Appendix B.

4.1 Translation Quality

Following the reference-based automatic evaluation settings proposed by the GenMT task, translation quality is automatically evaluated using XCOMET-XL[8] (Guerreiro et al., 2024) and MetricX-24-Hybrid-XL[9] (Juraska et al., 2024). Results are shown in Table 3, in which we also report the number of potentially hallucinated lines for each system, providing insight into system robustness across evaluation conditions. An output line was considered hallucinated if its output length (measured as number of characters) was at least twice the length of the source. Since we did not observe significant score differences across decoding with various batch sizes, we only report metric scores computed from outputs generated with batch size 1. However, we did observe minor differences in the outputs at different batch sizes, which indicates that the submitted implementations do not account for padding handling e.g., for relative positions (Papi et al., 2024).

The main observation is that BitsAndBytes and LeanAya achieve compression with minimal quality loss, nearly matching the baseline with ≈55 COMET scores. Other compression methods can decrease quality significantly, in particular for Czech→German, with ≈31–44 COMET scores.

4.2 Model Size

For each system variant, we record model size as the on-disk footprint of the submitted model direc-

[8] Unbabel/XCOMET-XL
[9] google/metricx-24-hybrid-xl-v2p6

Submission	System	Quality		Size↓	Speed↑ (tok/s)		
		COMET↑	MetricX↓	(GB)	b=1	b=16	b=64
Baseline	base	55.3	5.08	15.0	33.3	153.6	170.2
BitsAndBytes	**bnb-8bit**	55.6	5.08	8.5	12.3	97.3	196.4
	bnb-4bit-fp4	54.5	5.24	5.6	23.2	63.2	93.2
	bnb-4bit-nf4	55.7	5.15	5.6	23.4	78.4	134.3
	bnb-4bit-nf4-2q	55.5	5.15	5.3	19.8	76.8	131.6
AyaQ	**mcptqsr**	40.3	8.70	7.6	48.5	165.5	264.1
LeanAya	**mcsr**	53.2	5.36	7.2	2.9	21.3	27.9
TCD-Kreasof	**primary**	39.9	7.93	12.0	42.7	19.3	–
	contrastive1	32.4	9.49	11.0	50.8	22.9	–
	contrastive2	21.4	14.53	8.5	59.8	30.1	11.8
Vicomtech	gelaco-0.25_ft_q4	41.2	7.52	4.5	28.6	111.4	229.1
	gelaco-0.25_ft_q8	44.4	6.75	6.9	14.5	77.5	171.7
	gelaco-0.50_ft_q4	31.0	9.82	3.7	40.9	125.5	271.5
	gelaco-0.50_ft_q8	33.7	9.24	5.3	21.8	81.1	180.2
	gelaco-0.75_gkd	30.6	11.03	7.2	129.0	366.5	636.8
	gelaco-0.75_gkd_q4	31.1	11.04	2.9	68.3	181.9	366.5
	gelaco-0.75_gkd_q8	31.1	10.82	3.6	39.0	102.5	204.3

Table 4: Final results of the WMT25 Model Compression shared task. Primary submission names are bolded.

tory. The size is the sum of parameter shard files, tokenizer, and minimal wrapper scripts, which directly reflects the storage and transfer cost of deploying the model in gigabytes (GB).

We do not measure peak CPU memory usage or GPU VRAM footprint.

The model sizes are reported in Table 4. Organizer's submission BitsAndBytes (4-bit and 8-bit), while maintaining the translation quality, reduces the model size by up to 65%. On the other hand, Vicomtech's systems achieve best compression (81%, down to 2.9 GB from 15.0 GB) but suffer significant quality degradation.

4.3 Inference Speed

Each model was run three times per batch size, and wall-clock time was recorded for each run. Our primary metric, output rate, is defined as the number of output tokens divided by adjusted wall time (tokens/s). Output tokens were counted by retokenizing the generated hypotheses using the Aya Expanse 8B tokenizer. To isolate model initialization overhead, we performed a separate "warmup" run per model, decoding a single short sentence with batch size 1. The average wall time of warmup runs was subtracted from the total wall time to compute an adjusted speed metric. We also tested multiple batch sizes to analyze throughput scaling. The results for primary systems are presented on Figure 1.

As expected, inference speed scales with batch size, but not uniformly. Vicomtech's systems scaled most efficiently from 70 tokens/s at batch size 1 to nearly 1000 tokens/s at 512. Some models saturated early, showing minimal speedup with larger batch sizes or even failing to produce outputs.

5 Conclusion and Future Directions

The final results of the WMT25 shared task on model compression are summarized in Table 4. Figure 2 shows Pareto front comparisons across evaluation criteria.

The key findings can be summarized as follows:

- BitsAndBytes baselines and LeanAya maintained translation quality with moderate speed and model size reduction;
- Vicomtech's systems achieved best compression rates, latency and throughput thanks to efficient batch scaling, but at the cost of translation quality;
- Quantization has emerged as the most popular approach for its simplicity and effectiveness;
- Hallucinations in compressed outputs reveal the fragility of the current approaches and the need for more robust evaluation and compression-aware training techniques.

Overall, despite the moderate participation in this shared task limited the breadth of exploration, several submissions showed promising results. The results of this evaluation campaign highlight that task-specific compression of LLMs still warrants

more research efforts, especially at high compression rates required for running systems on edge devices. The smallest model still requires almost 3GB of disk space, which is incompatible with many edge devices that are equipped with a few hundred MB of memory (Cai et al., 2022). Additionally, high compression rates result in a significant performance drop. We believe that pushing the boundaries of compression rates and reducing the quality degradation in such settings represent the most interesting challenges for future research on the topic and for participants of the future editions of the task.

Looking ahead, future iterations of this task could benefit from expanding the evaluation to more language pairs, aligning more tightly with the evaluation benchmark at the GenMT task, and including human assessments of the outputs.

6 Limitations

This study offers an early glimpse into the landscape of model compression for machine translation, but several limitations constrain the generality of its findings. First, the participation was only modest, all systems compressed the same base model (Aya Expanse 8B) and primarily focused on one language pair (Czech→German). Second, the submissions relied mainly on quantization, with limited exploration of other well-established compression techniques such as parameter pruning or knowledge distillation. Third, only one system used vLLM infrastructure, limiting comparability.

Lastly, quality assessment depended solely on automatic metrics (COMET and MetricX). We did not conduct human evaluation or cross-language validation. We also did not present some important deployment metrics (e.g., memory usage, latency, energy consumption), which narrows the conclusions.

Acknowledgments

We would like to thank all participants for submitting their systems to the shared task. Marco Gaido's work has been funded by the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

References

Idris Abdulmumin, Victor Agostinelli, Tanel Alumäe, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Fethi Bougares, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, William Chen, Raj Dabre, Yannick Estève, Marcello Federico, Mark Fishel, Marco Gaido, Dávid Javorský, Marek Kasztelnik, Fortuné Kponou, Mateusz Krubiński, Tsz Kin Lam, Danni Liu, Evgeny Matusov, Chandresh Kumar Maurya, John P. McCrae, Salima Mdhaffar, Yasmin Moslem, Kenton Murray, Satoshi Nakamura, Matteo Negri, Jan Niehues, Atul Kr. Ojha, John E. Ortega, Sara Papi, Pavel Pecina, Peter Polák, Piotr Połeć, Ashwin Sankar, Beatrice Savoldi, Nivedita Sethiya, Claytone Sikasote, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Brian Thompson, Marco Turchi, Alex Waibel, Patrick Wilken, Rodolfo Zevallos, Vilém Zouhar, and Maike Züfle. 2025. Findings of the IWSLT 2025 evaluation campaign. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 412–481, Vienna, Austria (in-person and online). Association for Computational Linguistics.

Red Hat AI and vLLM Project. 2024. LLM Compressor.

Alexandra Birch, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Yusuke Oda. 2018. Findings of the second workshop on neural machine translation and generation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.

Han Cai, Ji Lin, Yujun Lin, Zhijian Liu, Haotian Tang, Hanrui Wang, Ligeng Zhu, and Song Han. 2022. Enable deep learning on mobile devices: Methods, systems, and applications. *ACM Trans. Des. Autom. Electron. Syst.*, 27(3).

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.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Jacob Devlin. 2017. Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the CPU. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2820–2825, Copenhagen, Denmark. Association for Computational Linguistics.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. 2020. Stabilizing the lottery ticket hypothesis.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. In *The Eleventh International Conference on Learning Representations*.

Thamme Gowda, Roman Grundkiewicz, Elijah Rippeth, Matt Post, and Marcin Junczys-Dowmunt. 2024. PyMarian: Fast neural machine translation and evaluation in python. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 328–335, Miami, Florida, USA. Association for Computational Linguistics.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. 2019. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong. Association for Computational Linguistics.

Kenneth Heafield, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li, and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 1–9, Online. Association for Computational Linguistics.

Kenneth Heafield, Biao Zhang, Graeme Nail, Jelmer Van Der Linde, and Nikolay Bogoychev. 2022. Findings of the WMT 2022 shared task on efficient translation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 100–108, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Kenneth Heafield, Qianqian Zhu, and Roman Grundkiewicz. 2021. Findings of the WMT 2021 shared task on efficient translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 639–651, Online. Association for Computational Linguistics.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

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

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Natalia Fedorova, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Howard Lakougna, Jessica Lundin, Kenton Murray, Masaaki Nagata, Stefano Perrella, Lorenzo Proietti, Martin Popel, Maja Popović, Parker Riley, Mariya Shmatova, Steinþór Steingrímsson, Lisa Yankovskaya, and Vilém Zouhar. 2025. Preliminary ranking of wmt25 general machine translation systems.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yasmin Moslem, Muhammad Hazim Al Farouq, and D. John Kelleher. 2025. Iterative Layer Pruning for Efficient Translation Inference. In *Proceedings of the Tenth Conference on Machine Translation*, Suzhou, China.

Sara Papi, Marco Gaido, Andrea Pilzer, and Matteo Negri. 2024. When good and reproducible results are a giant with feet of clay: The importance of software quality in NLP. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3657–3672, Bangkok, Thailand. Association for Computational Linguistics.

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

David Ponce, Harritxu Gete, and Thierry Etchegoyhen. 2025b. Vicomtech@WMT 2025: Evolutionary Model Compression for Machine Translation. In *Proceedings of the Tenth Conference on Machine Translation*, Suzhou, China.

Ricardo Rei, Nuno M. Guerreiro, JosÃ© Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.

Shicheng Tan, Weng Lam Tam, Yuanchun Wang, Wenwen Gong, Shu Zhao, Peng Zhang, and Jie Tang. 2023. GKD: A general knowledge distillation framework for large-scale pre-trained language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 134–148, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang and Anshumali Shrivastava. 2025. LeanQuant: Accurate and Scalable Large Language Model Quantization with Loss-error-aware Grid. In *International Conference on Representation Learning*, volume 2025, pages 35521–35544.

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

A Hallucinations

To understand if the potential hallucinations impacted the results, we benchmarked the participating systems on a subset of the Czech→German WMT25 test set. This subset includes only segments where none of the systems exhibits hallucinations under any batch setting. An output line was considered hallucinated if its tokenized output length was at least twice the length of the tokenized source. This version of the Czech-German test set reduces the number of input segments from 2,686 to 1,928 segments. Table 5 illustrates quality comparisons across systems for both versions of the test set using a reference-less metric, WMT23-CometKiwi-XL (Rei et al., 2023), computed using Pymarian (Gowda et al., 2024). Inference speed metrics across different settings are presented in Figures 3 and 4.

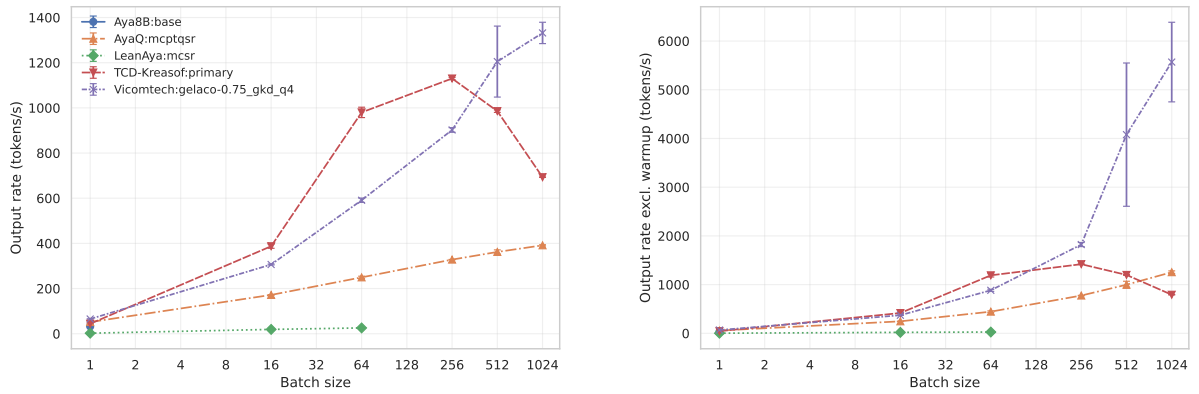
Submission	**System**	*full* **CKiwi↑**	*subset* **CKiwi↑**	**#Halluc.**
Baseline	base	66.9	71.1	83
BitsAndBytes	**bnb-8bit**	66.8	—	86
	bnb-4bit-fp4	66.5	70.8	153
	bnb-4bit-nf4	66.6	70.7	103
	bnb-4bit-nf4-2q	66.7	70.8	106
AyaQ	**mcptqsr**	58.9	64.9	200
LeanAya	**mcsr**	66.3	70.9	66
TCD-Kreasof	**primary**	59.7	64.4	78
	contrastive1	55.0	58.8	102
	contrastive2	39.3	42.9	335
Vicomtech	gelaco-0.25_ft_q4	60.5	64.1	37
	gelaco-0.25_ft_q8	61.8	65.4	42
	gelaco-0.50_ft_q4	53.3	56.7	94
	gelaco-0.50_ft_q8	55.9	59.3	52
	gelaco-0.75_gkd	54.8	59.3	198
	gelaco-0.75_gkd_q4	54.5	59.9	187
	gelaco-0.75_gkd_q8	55.2	60.6	197

Table 5: COMET-Kiwi-XL scores for the original Czech→German WMT25 test set and the filtered version without lines exhibiting potential hallucinations.

B Detailed results

Figure 5 presents extended evaluation of the average output token rates across multiple batch sizes for all submissions, including contrastive submissions.

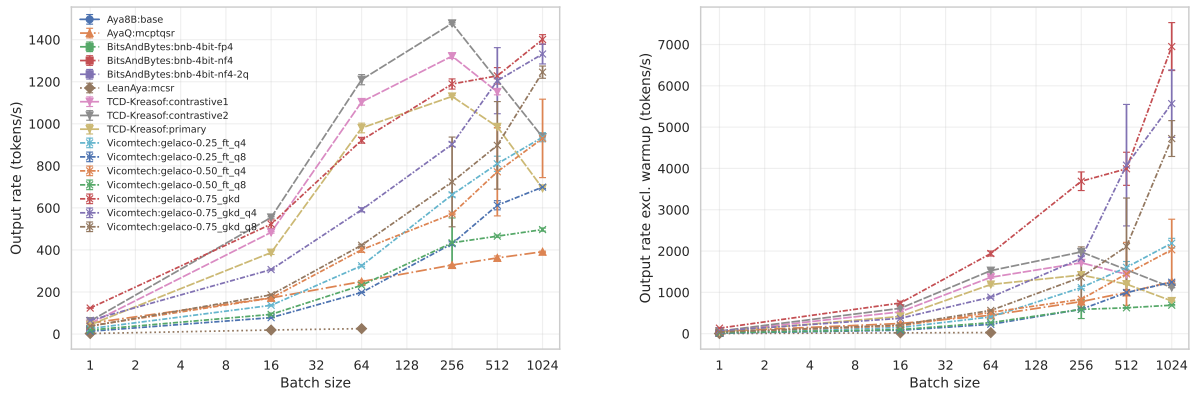
Table 6 provides details about warmup times and total decoding times for two batch size settings for each system.



(a) Total output rates.

(b) Output rates excluding warmup times.

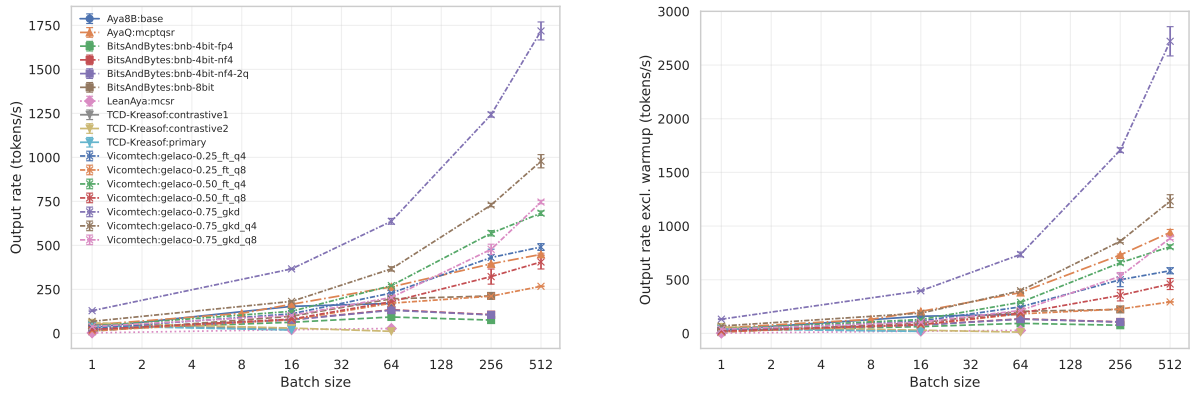
Figure 3: Inference speed as tokens/s when translating **the subset of the Czech→German WMT25 test set not causing hallucinations**. Mean and standard deviation across 3 runs, reported for various batch sizes. Primary submissions only.



(a) Total output rates.

(b) Output rates excluding warmup times.

Figure 4: Inference speed as tokens/s when translating **the subset of the Czech→German WMT25 test set not causing hallucinations**. Mean and standard deviation across 3 runs, reported for various batch sizes. Primary and contrastive submissions.



(a) Total output rates.

(b) Output rates excluding warmup times.

Figure 5: Inference speed as tokens/s when translating the entire Czech→German WMT25 test set. Mean and standard deviation across 3 runs, reported for various batch sizes. **Primary and contrastive submissions**.

Submission	System	Warmup (sec.)	Time (sec.)	Speed↑ (tok/s) total	excl.w.	Time (sec.)	Speed↑ (tok/s) total	excl.w.
			Batch size 1			Batch size 16		
Baseline	base	22.5	2,659.3	33.3	33.6	576.6	153.6	159.9
BitsAndBytes	**bnb-8bit**	19.7	7,231.4	12.3	12.3	923.2	97.3	99.4
	bnb-4bit-fp4	7.6	4,065.9	23.2	23.2	1,492.0	63.2	63.5
	bnb-4bit-nf4	7.4	3,762.6	23.4	23.5	1,123.1	78.4	78.9
	bnb-4bit-nf4-2q	7.8	4,461.4	19.8	19.8	1,148.7	76.8	77.3
AyaQ	**mcptqsr**	62.5	1,111.5	48.5	51.4	325.9	165.5	204.8
LeanAya	**mcsr**	25.7	32,198.5	2.9	2.9	4,291.8	21.3	21.4
TCD-Kreasof	**primary**	10.1	4,228.0	42.7	42.8	8,931.0	19.3	19.3
	contrastive1	9.2	4,033.2	50.8	50.9	8,970.6	22.9	22.9
	contrastive2	8.5	2,994.1	59.8	60.0	5,412.6	30.1	30.2
Vicomtech	gelaco-0.25_ft_q4	25.6	2,711.9	28.6	28.9	729.6	111.4	115.4
	gelaco-0.25_ft_q8	26.4	5,296.0	14.5	14.6	1,015.7	77.5	79.6
	gelaco-0.50_ft_q4	25.6	2,683.6	40.9	41.3	833.1	125.5	129.5
	gelaco-0.50_ft_q8	25.3	4,083.0	21.8	22.0	1,155.5	81.1	83.0
	gelaco-0.75_gkd	26.0	942.1	129.0	132.7	339.2	366.5	396.9
	gelaco-0.75_gkd_q4	25.4	1,773.0	68.3	69.3	670.6	181.9	189.1
	gelaco-0.75_gkd_q8	26.4	3,201.9	39.0	39.3	1,201.4	102.5	104.8

(a) Speed metrics for **batch sizes 1 and 16**.

Submission	System	Warmup (sec.)	Time (sec.)	Speed↑ (tok/s) total	excl.w.	Time (sec.)	Speed↑ (tok/s) total	excl.w.
			Batch size 64			Batch size 256		
Baseline	base	22.5	520.5	170.2	177.9	–	–	–
BitsAndBytes	**bnb-8bit**	19.7	456.6	196.4	205.3	419.1	213.0	223.5
	bnb-4bit-fp4	7.6	1,011.3	93.2	93.9	1,245.5	75.6	76.1
	bnb-4bit-nf4	7.4	655.5	134.3	135.8	825.8	106.6	107.6
	bnb-4bit-nf4-2q	7.8	670.3	131.6	133.1	839.5	105.0	106.0
AyaQ	**mcptqsr**	62.5	204.0	264.1	380.8	136.0	395.2	731.0
LeanAya	**mcsr**	25.7	3,293.0	27.9	28.1	–	–	–
TCD-Kreasof	**primary**	10.1	–	–	–	–	–	–
	contrastive1	9.2	–	–	–	–	–	–
	contrastive2	8.5	12,428.5	11.8	11.8	–	–	–
Vicomtech	gelaco-0.25_ft_q4	25.6	346.0	229.1	247.4	188.6	431.1	500.2
	gelaco-0.25_ft_q8	26.4	455.9	171.7	182.2	384.7	211.5	227.1
	gelaco-0.50_ft_q4	25.6	403.0	271.5	290.0	188.0	569.0	658.7
	gelaco-0.50_ft_q8	25.3	518.0	180.2	189.5	288.5	322.8	354.9
	gelaco-0.75_gkd	26.0	193.8	636.8	735.7	95.6	1,242.6	1,707.5
	gelaco-0.75_gkd_q4	25.4	327.3	366.5	397.4	168.5	729.3	858.9
	gelaco-0.75_gkd_q8	26.4	582.1	204.3	214.0	255.6	477.5	532.6

(b) Speed metrics for **batch sizes 64 and 256**.

Table 6: Detailed speed metrics including the total translation time of the Czech→German WMT25 test set, the total token output rate and token output rate excluding warmup. Averages across 3 runs for various batch sizes.