

One Model, Five Tasks, Two RTX 3090 GPUs: The HSE System for the Sorbian Track of WMT26 Multitask LLMs with Limited Resources

Michael Diskin
HSE University
Moscow, Russia
michael.s.diskin@gmail.com

Abstract

We describe HSE’s submission to the Sorbian track of WMT26 Multitask LLMs with Limited Resources, where a single 2B-parameter model must translate between German, Upper Sorbian, and Lower Sorbian and also answer multiple-choice questions, check spelling and grammar, and solve maths problems in both Sorbian languages. Our system is one LoRA adapter on Qwen3.5-2B, trained on the official parallel data, synthetic spelling and grammar examples, a question-answering proxy, and English GSM8K in about nine GPU-hours on two consumer RTX 3090 GPUs. Extended translation-only fine-tuning cost about ten points of question-answering accuracy, reproducing a WMT25 finding, and nearly erased the output conventions of the other tasks; the multitask mixture recovered them at almost no translation cost. The system ranked fourth of four teams. It far exceeded the baseline on translation (21.7 $\rightarrow$ 61.5 chrF++) and spell checking (6.7 $\rightarrow$ 63.6), but on grammar checking and maths reasoning it learned the answer format without the skill: the grammar score is almost exactly the share of error-free sentences, and maths accuracy did not exceed the baseline. Two negative results may transfer to other low-resource settings. Synthetic holdouts did not predict performance on the official tasks in either direction, and self-translated training problems passed every automatic check yet changed meaning in at least 6 of 20 inspected cases. The model is publicly available.

1 Introduction

The WMT26 Multitask LLMs with Limited Resources shared task (Okabe et al., 2026), a successor to the WMT25 shared task on low-resource Slavic languages (Okabe et al., 2025), requires each participant to submit *one* model that handles five tasks. The Sorbian track comprises Machine Translation (MT) in six directions (de $\leftrightarrow$ hsb, de $\leftrightarrow$ dsb,

and hsb $\leftrightarrow$ dsb), Multiple-Choice Question Answering (QA), Spell Checking (SC), Grammar Checking (GC), and Maths Reasoning (MR) in Upper and Lower Sorbian. Upper Sorbian (approximately 30,000 speakers) and, especially, Lower Sorbian (approximately 7,000 speakers) are extremely low-resource languages, while the final ranking weights all five tasks equally. Eligible models are restricted to the 2B Qwen3.5 family (Qwen Team, 2026).

Our system is a single LoRA adapter (Hu et al., 2022) on Qwen3.5-2B, trained on official parallel data, synthetic task examples, and English GSM8K within the budget of two consumer GPUs. We quantify the trade-off between MT and the other four tasks that motivates the multitask mixture (§4), show where the recipe works and where the model learns only the answer format (§5), and report why two shortcuts failed: synthetic holdouts as a selection signal and self-translated training problems. Along the way we revisit two WMT25 observations: we reproduce the MT–QA trade-off and identify a lighter training regime in which it did not appear, and our self-translation pilot points to a second possible mechanism behind the negative result that JGU Mainz reported for translated training questions (Saadi et al., 2025). We release the model¹ and document its training and evaluation.

2 Related Work

Sorbian MT and multitask adaptation. The WMT20–22 shared tasks covered translation between German, Upper Sorbian, and Lower Sorbian (Fraser, 2020; Libovický and Fraser, 2021; Weller-Di Marco and Fraser, 2022). WMT25 evaluated MT and QA jointly for low-resource Slavic languages (Okabe et al., 2025). Its systems exposed a tension between the two tasks: NRC’s MT-only system fell below random performance on Upper

¹https://huggingface.co/yhn112/WMT_Limited_Resources_Sorbian

Sorbian QA (Larkin et al., 2025), while TartuNLP found that a separate MT fine-tuning stage impaired QA (Purason and Fishel, 2025). TartuNLP used a single-stage multitask mixture (Purason and Fishel, 2025), and JGU Mainz retained parallel data during fine-tuning to prevent forgetting of MT (Saadi et al., 2025). Our system follows this multitask approach in the five-task WMT26 setting.

Synthetic data for error correction. Synthetic error generation has been used to train grammatical error correction systems, with downstream quality depending on both the corruption process and the training schedule (Grundkiewicz et al., 2019; Kiyono et al., 2019). Synthetic spelling errors have also been used for spelling correction in resource-scarce Indic languages (Etoori et al., 2018) and Slovene, where training-data construction strongly affected performance (Klemen et al., 2024). Our SC and GC experiments examine both the usefulness of synthetic training examples and the ability of synthetic holdouts to predict performance on the official tasks.

Translated supervision and mathematical reasoning. Machine-translated training data have been used for cross-lingual QA (Sorokin et al., 2022) and multilingual mathematical reasoning (Chen et al., 2024). For Sorbian, JGU Mainz translated about 200,000 English multiple-choice questions with its own models; fine-tuning on them slightly reduced QA accuracy, which the authors attribute to the joint MT/QA trade-off (Saadi et al., 2025). The reliability of translated reasoning data remains a concern: Zhu et al. (2024) documented errors in translated GSM8K reasoning traces. Our self-translation pilot examines a related risk in the problem statements themselves: changes in meaning that are missed by checks of answer tags and numerical content. Beyond translation, prior work has explored training on model-generated rationales selected by answer correctness (Zelikman et al., 2022) and synthetic proofs with automated quality checks (Obozov et al., 2025). English-pivoted chain-of-thought training, with reasoning in English and answers in the target language, has also improved mathematical reasoning in Irish (Tran et al., 2026). Our GSM8K supervision instead uses answer-only targets (§5).

Component	Rows	Weight	Token share
MT (official parallel)	234,594	1	81.8%
SC (synthetic)	15,000	1	3.6%
GC (synthetic)	1,800	10	3.9%
MR (English GSM8K)	7,094	6	5.4%
QA proxy	20,000	12	5.4%
Total	278,488	—	100%

Table 1: The multitask training mixture. Weight is the multiplier applied in the weighted cross-entropy objective; token share is the proportion of weighted supervised tokens.

3 Task, Data, and Constraints

Tasks and metrics. MT is evaluated in six directions using chrF++ (Popović, 2017), with BLEU (Post, 2018) also reported. QA uses accuracy; SC and GC use the mean of detection and correction exact match, including sentences with no error; MR uses answer exact match. We report all task scores on a 0–100 scale. The Sorbian MR evaluation uses PluraMath (Dementieva et al., 2026), an extension of PolyMath to additional languages. The organizers also evaluate the submitted model on a hidden QA set.

We generated all submitted outputs with the unchanged official lm-evaluation-harness fork (Gao et al., 2026), so that our inference configuration matches the baseline’s; the task does not require this. Generative answers are extracted from tags; QA is scored by multiple-choice log likelihood without a chat template, and MR is decoded by sampling at temperature 1. Appendix C gives the prompts, submission format, and stopping diagnostics.

Data restrictions. Participants may use any publicly available data but no evaluation-set material in any form, including translations: the Sorbian language-certificate exam questions that form the QA task, PolyMath for MR, and the previous year’s test sets.

4 Final System

Architecture and training. The base model is Qwen3.5-2B (1.9 billion parameters). We add a single LoRA adapter with rank 16, $\alpha = 32$, dropout 0.05, applied to all linear layers (Mantrkar et al., 2022). We use supervised fine-tuning with weighted cross-entropy, a maximum sequence length of 512, and a global batch size of 32. The initial training lasts two epochs, with a peak learn-

Model	MT	QA	SC	GC
Qwen3.5-2B base	22.2	49.3	9.4	1.7
MT only	63.1	38.7	3.2	2.2
Multitask (2 epochs)	62.9	51.6	61.6	50.0
Final (3 epochs)	64.2	53.4	63.9	49.9

Table 2: Official development results. MT is macro- chrF++ over the six directions; QA is accuracy; SC and GC are the mean of detection and correction exact match. All scores use a 0–100 scale. MR is omitted: with 24 examples per language and sampled decoding (§3), its development scores are too noisy to compare.

ing rate of $2 \cdot 10^{-4}$ and a cosine schedule. Training configuration and resource measurements are given in §6 and Appendix D.

Why a multitask mixture. Two development-set observations motivated the mixture (Table 2). First, extended MT-only training (the MT-only row of Table 2) reduced QA accuracy from 49 to 39, reproducing a central WMT25 finding (Okabe et al., 2025). Second—a dimension not present at WMT25—MT training nearly erased the *output conventions* of the other tasks. Even a lighter MT-only adapter, trained for one epoch on balanced parallel data (revisited in §5), reduced SC from 9 to approximately 3 and the valid-tag rate to 68–82%, while no-error answers almost disappeared: the model began to *rewrite* the sentence as if translating it. Multitask mixing was already standard at WMT25 (§2); here we quantify the trade-off across five tasks. Relative to MT-only training, the mixture gives up less than half a chrF++ point on MT while recovering 13 points on QA, 58 on SC, and 48 on GC.

Mixture composition. Table 1 gives the composition. All synthetic data were generated from the parallel training corpus, screened against the released development and test texts, and split by source group before generation; the held-out source groups were also removed from the MT component (Appendix B). The synthetic and external components are:

- **SC:** parallel-corpus sentences with a misspelling introduced into one word by a single character edit (diacritic substitution, insertion, deletion, or transposition; half of the examples contain no error), represented in the exact answer format of the task;
- **GC:** 1,800 examples with controlled subject–auxiliary agreement errors, the only class of

grammatical phenomena that we could generate confidently from the parallel data alone; this restriction had the largest effect on the final outcome (§5);

- **QA proxy:** multiple-choice translation-selection and lexical cloze examples generated from parallel sentence pairs; and
- **MR:** the English GSM8K training set (Cobbe et al., 2021) with answer-only targets: the assistant turn contains the final number in the task’s answer tags and no solution steps. Translating the questions into Sorbian with an internal MT model was rejected on quality grounds (§5).

An LLM agent, not a native speaker, reviewed 48 examples per synthetic family against the source sentence pairs, and all passed; Appendix B gives the construction rules, splits, leakage screens, and review protocol.

Continued training. We then continued training the same adapter for one epoch on the unchanged mixture at a learning rate of $5 \cdot 10^{-5}$ and selected the checkpoint at the end of the epoch on the official development set. It improved MT macro- chrF++ by 1.3 points and produced small gains on QA and SC while leaving GC unchanged (Table 2); the gain also transferred to the submission leaderboard, +1.6 BLEU and +1.2 chrF++ over the two-epoch model (§5). Returns were diminishing, so we stopped after one additional epoch.

5 Results and Analysis

Table 3 presents the official final results (Okabe et al., 2026), and Table 4 breaks down MT by direction. Development scores overestimated test scores but preserved their ordering. For the final model, development macro- chrF++ was 64.2 against 61.5 on the official test set. For three successive submissions, development macro- chrF++ rose from 56.7 to 62.9 to 64.2, and the leaderboard chrF++ of the same models rose in step from 54.1 to 59.5 to 60.7.² Leaderboard scores were never used to choose data, checkpoints, or inference settings. We therefore treat the development set as an optimistic but usable proxy for checkpoint selection.

²Leaderboard scores are corpus-level chrF++ over the concatenated output of all six directions, whereas the development and official final scores are means across directions, so levels are not directly comparable between the two.

Team	MT (chrF++)	QA	SC	GC	MR	Points	Rank
HeyBusan	72.75	55.92	78.03	79.52	29.00	22	1
LT3	69.57	65.05	76.11	78.32	30.40	22	1
Hitsz	63.12	57.33	73.13	62.11	18.60	16	3
HSE (ours)	61.50	46.77	63.63	50.02	5.00	9	4
baseline	21.68	43.39	6.66	1.34	5.40	5	5

Table 3: Official final results for the Sorbian track, using each team’s primary system; QA includes the hidden set. All task scores use a 0–100 scale. Bold denotes the best result for each task.

Direction	BLEU	chrF++	Best chrF++
de→hsb	30.48	56.27	70.26
hsb→de	27.14	52.70	69.43
de→dsb	27.14	52.72	63.06
dsb→de	29.90	54.61	69.42
dsb→hsb	55.78	75.50	82.38
hsb→dsb	57.32	77.19	81.96

Table 4: Our system’s final-test MT results by direction, with the best track chrF++ score (HeyBusan) for comparison.

Overall performance. The gains over the baseline are concentrated in MT and SC (Table 3); the apparent GC gain is examined below. The gap to the leaders was smallest for MT (11 chrF++ points overall and only 5–7 points for the bridge directions hsb↔dsb) and largest for MR and for GC on error-containing sentences.

QA: the WMT25 MT–QA trade-off appeared only under intensive MT training. The ten-point QA drop under MT-only training (§4) appeared only under the more intensive configuration of two epochs at the peak learning rate. The lighter MT-only adapter of §4, trained for one epoch on balanced parallel data at half the peak learning rate, did not degrade QA: hsb was unchanged, while dsb improved by approximately 6 points. Because the two conditions also differed in data volume and batch settings, we do not attribute the drop to any single factor. Interpolating the MT-only and multitask adapters recovered part of the MT and QA difference but lost some of the few correct GC predictions on sentences containing errors (Table 5). We kept the multitask model.

SC: synthetic data were effective. The system learned to detect and correct misspellings from synthetic edits. On the test set, detection/correction scores were 72/53 for hsb and 73/57 for dsb: correction remained harder than detection in both languages.

GC: almost all accuracy comes from the no-error half. Our final GC score was approximately 50. In the test package, the model returned *no error* for 3,712 of the 3,726 GC rows. The narrow synthetic agreement data taught the output format but not how to identify real errors, and the final score is almost exactly the proportion of error-free sentences. The same collapse was visible on the development set, where performance on the error-containing subset was close to zero. By contrast, the two-epoch model achieved 98.5% paired exact match (both fields correct) on the synthetic holdout, which failed to predict transfer.

MR: correct formatting without correct answers. The final MR score was 5.0, compared with 5.4 for the baseline: transfer from English GSM8K to Sorbian MR was negligible, and the two alternatives discussed below, the synthetic continuation and self-translated problems, had drawbacks of their own. Because the GSM8K targets contained only the final answer, the model was never trained to write a solution before answering; our experiments do not separate this target choice from the language-transfer problem. In a later run whose GSM8K targets wrapped the answer in TeX dollar delimiters, the LaTeX form that the official prompt invites (Table 5, stopping interventions), answers in that form rose from 2 to 45 of 48 problems, while correct answers rose only from 0 to at most 2.

Synthetic holdouts: unreliable predictors of task performance. We tried to improve GC and MR by briefly continuing training on new synthetic data: a morphology-grounded GC generator covering case, gender, number, and person, which yielded only 634 examples, and template-generated Sorbian MR problems, mixed with replay of the original tasks (Appendix B). Selecting the checkpoint by its synthetic-holdout scores picked one that nearly solved the holdouts but lost about 12 SC and 11 GC points on the official development set (Table 5).

After the final submission, we observed the reverse mismatch in SC-only models trained in rounds: after each round, the adapter was merged into the model and a new one was trained, resembling ReLoRA and Chain-of-LoRA (Lialin et al., 2024; Xia et al., 2024). They improved on the final model’s official development score of 63.9, reaching 74.0, while all three achieved 80–81% paired exact match on the source-disjoint synthetic holdout. The later evaluation covered SC alone, so retention of the other four tasks was untested and we did not replace the unified model. Together, these experiments show that a narrow synthetic evaluation can both reward changes that fail to transfer and miss improvements on the target task.

Self-translation: structural checks miss semantic errors. Our pilot suggests a second mechanism behind the JGU Mainz result (§2). We translated 100 GSM8K problems into both languages with an internal diagnostic MT model (Appendix A). Every translation passed the automatic checks of well-formed tags and matching numbers, yet at least 6 of the 20 manually inspected source pairs had altered semantics (*twice*→*once*, loss of the question, or a change of referent). For low-resource languages, where large training corpora cannot readily be proofread by native speakers, this pilot supports caution about self-translated training data—especially reasoning data—even when the MT model passes structural checks.

6 Efficiency on Consumer Hardware

The final model cost about 6 GPU-hours for the initial two epochs and about 3 for the additional epoch, and the entire experimental trajectory described in this paper, including rejected branches and evaluations, ran on the same two-GPU consumer server. The main trainer changes behind this budget were disabling gradient checkpointing, disabling unused-parameter detection in distributed data parallel training, and compiling the model with `torch.compile`; the production run processed about 52 examples per second at full sequence length and full logits. Two standard optimizations did not pay off for this model. Liger fused cross-entropy (Hsu et al., 2025) saved about 3 GiB but nearly halved throughput, because the 248,000-token vocabulary splits each short step into dozens of small language-model-head chunks, and FlashAttention-2 (Dao, 2024) was slower than PyTorch SDPA on our short sequences. Every

adopted change preserved numerical equivalence. Appendix D gives the individual measurements, memory checks, and the power-limit comparison.

7 Conclusion

For a five-task track with one 2B model, a single multitask LoRA adapter on public data was enough to beat the baseline by a wide margin on MT and SC within a consumer-GPU budget, but not to acquire the skills behind GC and MR, whose absence coincides with the lack of permitted in-domain training data. The remaining gap is therefore one of skill rather than format: closing it requires training signal for MR and for error-containing GC, and stronger non-bridge MT. Our two failed shortcuts argue for checking transfer on the official tasks early instead of trusting synthetic holdouts or automatic checks of translated data.

Limitations

Scope of the claims. The largest shortcomings are MR and GC on error-containing sentences. Both coincide with the absence of permitted, high-quality in-domain training data, but our experiments do not distinguish data coverage from optimization or model capacity.

Evidence quality. Because model selection used a single seed and small official development sets—158 QA questions and 24 MR examples per language, the latter decoded by sampling—we do not treat changes of 0.1–0.5 chrF++ or a few examples as statistically reliable. WMT25 teams also noted the instability of small QA sets (Larkin et al., 2025; Purason and Fishel, 2025). Synthetic data were reviewed with AI assistance but not by native speakers, so systematic generator errors may remain. LLM agents also prepared runs, collected metrics, and maintained logs; scientific decisions and external actions remained with the author.

Train–test source overlap. The MT component contains 763 records, about 0.3% and almost all in *de*→*dsb*, whose normalized sources also occur among the released MT test sources. These records stayed in the training data, neither removed nor upweighted, and the overlap list itself was not used for anything. We trained no model without them, so their effect on MT scores is unmeasured.

Inference configuration. We did not explore task-specific prompts, stop sequences, or decoding

settings. The model frequently continues generating after the first answer; the official parsers extract that answer, leaving track scores largely unaffected. Use outside the harness should stop at the closing tag. Appendix C reports the stopping diagnostics.

Acknowledgments

This research was supported by the Basic Research Program at HSE University (HSE-BR-2025-025).

We thank the track organizers for the data, evaluation harness, and timely feedback. We also thank Anastasiia Fadeeva for her moral support and invaluable help with the manuscript, and Dmitry Abulkhanov for insightful discussions.

References

- Jason Ansel, Edward Yang, Horace He, et al. 2024. [PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation](#). In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS 2024)*, pages 929–947, La Jolla, CA, USA. ACM.
- Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. 2024. [Breaking language barriers in multilingual mathematical reasoning: Insights and observations](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7001–7016, Miami, Florida, USA. Association for Computational Linguistics.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Tri Dao. 2024. [FlashAttention-2: Faster attention with better parallelism and work partitioning](#). In *International Conference on Learning Representations*.
- Daryna Dementieva, Nikolay Babakov, Kathy Hämerl, Ilseyar Alimova, Jindřich Libovický, Shu Okabe, Miras Baisbay, Lukas Edman, Abrorkhon Inomkhujaev, Antonia Karamolegkou, Mateusz Lango, Volkan Özer, Nikola Selic, Subhankar Swain, Tsendeniya Kinfe Temesgen, Galit Bary Weisberg, and Alexander Fraser. 2026. [PluraMath: Extending mathematical reasoning evaluation beyond high-resource languages](#). *Preprint*, arXiv:2607.05992.
- Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. 2018. [Automatic spelling correction for resource-scarce languages using deep learning](#). In *Proceedings of ACL 2018, Student Research Workshop*, pages 146–152, Melbourne, Australia. Association for Computational Linguistics.
- Alexander Fraser. 2020. [Findings of the WMT 2020 shared tasks in unsupervised MT and very low resource supervised MT](#). In *Proceedings of the Fifth Conference on Machine Translation*, pages 765–771, Online. Association for Computational Linguistics.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2026. [The language model evaluation harness](#). Evaluation used the official shared-task fork: <https://github.com/TUM-NLP/llms-lim-res-eval-2026>.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. [Neural grammatical error correction systems with unsupervised pre-training on synthetic data](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.
- Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, Yanning Chen, and Zhipeng Wang. 2025. [Liger-Kernel: Efficient Triton kernels for LLM training](#). In *Championing Open-source Development in ML Workshop @ ICML25*. Tech report: arXiv:2410.10989.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. [An empirical study of incorporating pseudo data into grammatical error correction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.
- Matej Klemen, Martin Božič, Špela Arhar Holdt, and Marko Robnik-Šikonja. 2024. [Neural spell-checker: Beyond words with synthetic data generation](#). In *Text, Speech, and Dialogue (TSD 2024)*, Lecture Notes in Computer Science, pages 85–96. Springer.
- Samuel Larkin, Chi-Kiu Lo, and Rebecca Knowles. 2025. [NRC systems for the WMT2025-LRSL shared task](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 1133–1142, Suzhou, China. Association for Computational Linguistics.
- Vladislav Lialin, Sherin Muckatira, Namrata Shiva-gunde, and Anna Rumshisky. 2024. [ReLoRA: High-rank training through low-rank updates](#). In *International Conference on Learning Representations*.

- Jindřich Libovický and Alexander Fraser. 2021. [Findings of the WMT 2021 shared tasks in unsupervised MT and very low resource supervised MT](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.
- Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. <https://github.com/huggingface/peft>.
- Mark Obozov, Michael Diskin, Aleksandr Beznosikov, Alexander Gasnikov, and Serguei Barannikov. 2025. [Synthetic proofs with tool-integrated reasoning: Contrastive alignment for LLM mathematics with lean](#). In *Proceedings of The 3rd Workshop on Mathematical Natural Language Processing (MathNLP 2025)*, pages 195–202, Suzhou, China. Association for Computational Linguistics.
- Shu Okabe, Daryna Dementieva, Marion Di Marco, Lukas Edman, Katharina Haemmerl, Marko Měškank, Anita Hendrichowa, and Alexander Fraser. 2025. [Findings of the WMT 2025 shared task LLMs with limited resources for Slavic languages: MT and QA](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 503–519, Suzhou, China. Association for Computational Linguistics.
- Shu Okabe, Marion Di Marco, Kathy Hämmel, Daryna Dementieva, Lukas Edman, Marko Měškank, Anita Hendrichowa, and Alexander Fraser. 2026. Findings of the WMT26 Shared Task: Multitask LLMs with Limited Resources. In *Proceedings of the Eleventh Conference on Machine Translation*, Budapest, Hungary. Association for Computational Linguistics.
- Maja Popović. 2017. [chrF++: words helping character n-grams](#). In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Taido Purason and Mark Fishel. 2025. [TartuNLP at WMT25 LLMs with limited resources for Slavic languages shared task](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 1143–1150, Suzhou, China. Association for Computational Linguistics.
- Qwen Team. 2026. Qwen3.5: Towards native multimodal agents. <https://qwen.ai/blog?id=qwen3.5>. Model card: <https://huggingface.co/Qwen/Qwen3.5-2B>.
- Hossain Shaikh Saadi, Minh Duc Bui, Mario Sanz-Guerrero, and Katharina von der Wense. 2025. [JGU mainz’s submission to the WMT25 shared task on LLMs with limited resources for Slavic languages: MT and QA](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 1151–1157, Suzhou, China. Association for Computational Linguistics.
- Nikita Sorokin, Dmitry Abulkhanov, Irina Piontkovskaya, and Valentin Malykh. 2022. [Ask me anything in your native language](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 395–406, Seattle, United States. Association for Computational Linguistics.
- Khanh-Tung Tran, Barry O’Sullivan, and Hoang D. Nguyen. 2026. [Reasoning transfer for an extremely low-resource and endangered language: Bridging languages through sample-efficient language understanding](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):33277–33286.
- Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. [Neural text generation with unlikelihood training](#). In *International Conference on Learning Representations*.
- Marion Weller-Di Marco and Alexander Fraser. 2022. [Findings of the WMT 2022 shared tasks in unsupervised MT and very low resource supervised MT](#). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 801–805, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Wenhan Xia, Chengwei Qin, and Elad Hazan. 2024. [Chain of LoRA: Efficient fine-tuning of language models via residual learning](#). *Preprint*, arXiv:2401.04151.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. [STaR: Bootstrapping reasoning with reasoning](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.
- Wenhao Zhu, Shujian Huang, Fei Yuan, Shuaijie She, Jiajun Chen, and Alexandra Birch. 2024. [Question translation training for better multilingual reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8411–8423, Bangkok, Thailand. Association for Computational Linguistics.

A Additional Experiments

Several experiments in Table 5 start from the *merged final model*, the final adapter merged into the base weights, and train a fresh adapter on top. Appendix C discusses the stopping behaviour.

A.1 Settings and Detailed Measurements

Synthetic continuation. The 9,600-row mixture contained the 634 morphology-grounded GC ex-

Experiment	Outcome
Synthetic continuation Two-epoch multitask model; combined synthetic GC/MR data and five-task replay	The checkpoint selected on synthetic holdouts before official evaluation nearly solved those holdouts, but lost 12.4 SC and 11.1 GC points on the official development set.
Self-translation 100 GSM8K problems translated into both languages by a diagnostic MT model	All 200 translations had well-formed tags and matching numbers, yet at least 6 of the first 20 manually inspected source pairs had semantic errors (§5). We retained the English MR data.
Adapter interpolation Exact linear blends of the two-epoch multitask and MT-only LoRA updates	All three blends improved MT on a fixed development slice and QA, but each lost at least one of the few correct GC predictions on error-containing sentences. A rule fixed before the screen allowed no such regression, so we rejected all three on evidence of a handful of rows.
Stopping interventions (a) Otherwise identical runs with end-of-turn token weights 1 and 4, MR targets with the answer in TeX dollar delimiters in both arms; (b) sequence-unlikelihood penalty on post-tag continuations, with a plain cross-entropy control on the same data	(a) Weight 4 did not improve clean stopping over weight 1. Both arms raised dollar-delimited MR answers from 2/48 to 45/48 with at most 2/48 correct answers and lost 1.0–1.2 chrF++ relative to the merged final model. (b) Clean stopping rose from 23% to 67% on the probe’s evaluation slice, while MT lost 0.6 chrF++ and SC lost 4.3 points relative to the merged final model; relative to the control, the unlikelihood term itself cost 0.4 chrF++ and 0.4 SC points.
SC-only training Three merge-and-retrain rounds; source-disjoint synthetic holdouts	Official development SC rose monotonically from the final model’s 63.9 to 74.0, yet synthetic paired exact match plateaued at 80–81% and the synthetic screen rejected every round; the other four tasks were not evaluated.

Table 5: Additional experiments and their observed outcomes. Task score changes are in points on a 0–100 scale; stopping and paired exact match rates are percentages.

amples, a 1,366-example subset of the template-generated MR training set (Appendix B), and replay of the five tasks. Training lasted 300 steps at a learning rate of 10^{-5} ; the synthetic evaluation selected checkpoint 200.

Self-translation. Problems were stratified by length, number count, and answer form. The diagnostic MT model had been trained with official development data and was therefore ineligible for submission.

Adapter interpolation. With weights on the multitask LoRA update $\lambda = 0.80/0.90/0.95$, MT macro-chrF++ on a fixed 500-example development slice per direction improved by 1.1/0.8/0.2, and QA by 5.8/4.5/2.7 points, respectively, relative to the two-epoch multitask model. On error-containing GC, that model itself had only 3 detection and 2 correction exact matches among 1,000 hsb rows and none among 625 dsb rows; the blends retained at most 2 detection matches and no correction matches.

Stopping interventions. Both probes trained a fresh rank-16 adapter on the merged final model for 100 steps. The two weighting arms used the full

mixture at a learning rate of $5 \cdot 10^{-5}$ and differed only in the loss weight of the final `<|im_end|>` token. The unlikelihood run (Welleck et al., 2020) used 5,632 positive examples balanced over 11 generative task labels, each paired with one rejected continuation, at a learning rate of $2 \cdot 10^{-5}$.

SC-only training. Starting from the merged final model, we trained and merged a fresh rank-128 adapter in each round (one epoch, learning rate 10^{-4}). The first two rounds used 60,000 synthetic rows; the third added the previously evaluated 4,000-row holdout and was scored on a new source-disjoint holdout. Data included clean sentences and one- or two-character errors.

B Synthetic Data Construction and Review

All synthetic families were generated by deterministic local scripts from the released Sorbian MT training files (de-hsb, de-dsb, and hsb-dsb) and, for MR, from the English GSM8K training set. Released development and test rows entered the pipeline only through a collision screen that removes any candidate whose normalized text

matches a released development or test string; they never influenced selection. Each family assigned source rows to train or holdout before generation (20% holdout), and the source groups of every synthetic holdout were removed from the MT component, so the 4,779 holdout rows (2,000 SC, 400 GC, 2,000 QA, 379 MR) share no source sentence with training.

SC (15,000 train, 2,000 holdout). Source sentences pass length and cleanliness filters (5–35 words, terminal punctuation, no URLs or markup, no normalized duplicates). An error example alters one word by a single character using one of four families in equal shares: diacritic substitution, insertion, deletion, or transposition. The corrupted form must not occur in the training-corpus vocabulary of the language, and the source word must occur at least twice. Clean and error examples are balanced 1:1 and stratified by sentence length and error position; targets use the exact official answer format.

GC (1,800 train, 400 holdout). Source sentences contain exactly one adjacent pair of a nominative subject pronoun and a positive present-tense auxiliary. An error example replaces only the auxiliary with the form of another person/number cell; a clean example keeps the sentence. Each language contributes equal numbers of clean and error examples spread over the six agreement classes according to availability. Random word replacement, case and preposition errors, negative and future auxiliary paradigms, and sentences with zero or several controlled pairs were excluded.

QA proxy (20,000 train, 2,000 holdout). Two families in equal shares use the official raw multiple-choice format with four options and uniformly distributed correct positions. Translation selection presents a German sentence and asks for its Sorbian translation among distractors of similar length and punctuation from the same split. Lexical cloze asks for the correct word form among distractors that are known clean forms with the same initial letter, length ± 1 , and Levenshtein distance 1–3. Targets are the option number, scored by log likelihood without a chat template as in the official QA task.

MR (7,094 train, 379 holdout). GSM8K training problems were screened against the WMT26 MR development and test questions and split by source ID. Each target is the final numeric answer

inside the answer tags; the GSM8K solution text was not used.

Semantic review. For each family, a fixed stratified sample of 48 examples was reviewed row by row by an LLM agent with a trace to the German or paired-Sorbian source sentence. Acceptance required that the source is a usable sentence, that an error example contains exactly one intended error, and that the target is correct; for QA, that exactly one option is correct and the options are close and fluent. All 48 passed in every family. The only notes concerned loan words and lower-case sentence starts in the released sources, which led to a stricter source filter when the GC set was scaled up.

Holdout scoring. Holdouts were scored with the official tag extraction (SC, GC, MR) and raw multiple-choice mode (QA). These scores are checkpoint evidence only and are not comparable to official development scores.

Data for the synthetic continuation. Two further generators were built after the primary mixture was fixed and were used only in the short synthetic continuation (Table 5). (i) Morphology-grounded GC: from released sentences and two public morphology resources,³ an error example replaces exactly one surface form with an attested form of the same lemma and part of speech that differs in exactly one of case, gender, number, or person, and the replacement must break an explicit adjacent nominal or subject–verb agreement relation. The small Upper Sorbian lexicon (3,606 forms) limited the yield to 634 training and 180 holdout examples across both languages after an agent review of a 300-example sample. (ii) Template-generated Sorbian MR: arithmetic word problems instantiated from project-authored templates with paired numbers for both languages and solved by executing the underlying program. Training and holdout templates are disjoint, so the holdout measures wording transfer rather than memorized numbers. The continuation used a 1,366-example subset with targets reduced to the answer tags. No sentence from the MR evaluation data was used.

C Inference and Submission Format

A test submission comprises 34,815 rows: 26,174 for MT, 415 for QA, 4,000 for SC, 3,726 for GC,

³UD Upper Sorbian UFAL r2.18 (CC BY-SA 4.0) and UniMorph Lower Sorbian (CC BY-SA 3.0).

Translate the following German text to Upper Sorbian. Put it in this format
`<hsb> Upper Sorbian translation </hsb>.`
`<deu> {{de}} </deu>`

The following sentence contains at most one misspelled word. Respond with exactly these two tags and nothing else:
`<wrong> misspelled word </wrong>`
`<corrected> correct spelling </corrected>`
 If the sentence has no error, use the literal string CORRECT in both tags:
`<wrong> CORRECT </wrong>`
`<corrected> CORRECT </corrected>`
`<sentence> {{input_sentence}} </sentence>`

Answer this mathematical reasoning question. Place your final answer, using LaTeX when appropriate, in this format:
`<answer> answer </answer> {{question}}`

Figure 1: Official harness prompts (verbatim doc_to_text, including the original spelling; line breaks were added for typesetting): MT de→hsb, SC, and MR. The GC prompt replaces *misspelled word / correct spelling* with *grammatically incorrect word / correct form*. QA is scored by multiple-choice log likelihood on the raw prompt without a chat template. Answers are extracted from the tags using regular expressions.

and 500 for MR. QA is scored by log likelihood; the remaining 34,400 rows are generated. Figure 1 reproduces the official prompts.

Termination. Only 5,222 of the 34,400 generated rows in the final package stop cleanly at the closing tag. The parsers take the first tagged answer, so track metrics are largely unaffected, and a stop sequence at the closing tag would remove the symptom at inference time. Because we kept the official inference configuration unchanged (§3), we briefly tried two training-side fixes (Table 5): up-weighting the final end-of-turn token did nothing, while sequence-unlikelihood training on rejected continuations raised the clean-stop rate substantially at a small cost in MT and SC. Those rates were measured on the probes’ own evaluation slices and are not comparable to the package count above. Since a stop sequence achieves the same at inference time, we kept the primary model.

D Configuration Notes

Production training. Training used BF16, SDPA, torch.compile/Inductor (Ansel et al., 2024), and weighted cross-entropy over full logits. The global batch of 32 comprised a micro-batch of 8 per GPU, two GPUs, and two gradient-accumulation steps. The initial two epochs (17,406

steps) took approximately three hours on two RTX 3090 GPUs at 350 W; in the monitored part of the additional epoch (8,703 steps), reserved memory peaked at 17.0 GiB per card.

Trainer throughput. Throughput was measured after warm-up, separately from compilation time, and the longest batch was tested before each production run. On the eager trainer, disabling gradient checkpointing raised single-GPU throughput by 62%, disabling unused-parameter detection in DDP raised throughput by 29%, and gradient bucket views added 5%; the final eager trainer ran at 27.7 examples per second at batch 8, up from 17.8. On a short calibration corpus, torch.compile at batch 16 reached 80.7 examples per second, and the production run with full logits and length 512 reached 52.2. These settings differ in data and batch size and are not pairwise comparable.

Implementation comparisons. These configuration choices were tuned to our sequence-length profile using short probes that differed only in the setting under test. The production system used PyTorch 2.11/CUDA 12.8. The FlashAttention-2 and Liger comparisons are summarized in §6. The effect of the power limit depended on the batch setting: on the compiled batch-16 setup, increasing the limit from 250 to 350 W improved speed by 11%, whereas on the earlier eager setup it made no appreciable difference.