LoResMT 2026 Shared Task System Description

Vladimir Panov
Independent Researcher
vladimirpanov73@gmail.com

Abstract

We describe our submission to the shared task LoResMT 2026, which involved translating from low-resource Turkic languages Bashkir, Chuvash, Kazakh, Kyrgyz, and Tatar from English or Russian. We submitted runs for the English-Chuvash language pair using Neural machine translation (NMT). Our approach focused on systematic experimentation with diverse model architectures and an emphasis on optimizing inference-time parameters. The key findings indicate that a large-scale, specialized multilingual translation model, combined with targeted data preprocessing and careful generation tuning, yielded the best performance, achieving a chrF++ score of 29.67 on the public test set.

1 Introduction

The LoResMT 2026 Shared Task addresses the critical and underexplored problem of machine translation involving low-resource Turkic languages. This domain presents unique and formidable challenges, primarily stemming from the acute scarcity of high-quality parallel data. These limitations hinder the direct application of state-of-the-art methods that rely on massive datasets, necessitating innovative approaches tailored to data-constrained environments. This report details our approach, experiments, and results for the English to Chuvash translation task. We explore various neural machine translation (NMT) models, data preprocessing techniques, and training methodologies to improve translation quality for this under-resourced language pair. The main contributions of this work include a systematic comparison of different model families for English-Chuvash translation and a demonstration of the significant impact that inference-time parameter tuning can have on final translation quality.

2 Data

The shared task organizers provided a corpus for the Chuvash language, including a monolingual corpus and bilingual corpora for English-Chuvash (which was automatically aligned) and Russian-Chuvash. Additionally, data from the GATITOS dataset (Jones et al., 2023) and English (Latin script) and Chuvash (Cyrillic script) samples from the FLORES+ dataset (NLLB Team et al., 2024) were used.

Due to computational budget constraints and the exploratory nature of our initial experiments, we focused our training exclusively on the English-Chuvash corpus and the GATITOS dataset. We made a deliberate decision to exclude the substantially larger Russian-Chuvash parallel corpus and the extensive monolingual Chuvash data. Although these resources hold potential for future work (e.g., through back-translation), their inclusion at this stage would have led to prohibitive training times, making rapid iteration and model comparison impractical. The FLORES+ dataset was used to evaluate the trained model and to find the optimal generation parameters. Table 1 provides statistics for the datasets used in the training.

Dataset	# Sentences
English-Chuvash	204k
GATITOS en-cv	4k
FLORES+ (dev)	997

Table 1: Statistics of the primary datasets used for training and evaluation.

2.1 Data preprocessing

A multi-step pipeline was used for data preparation, including text cleaning, filtering examples with a large difference in character count between the source text and its translation, and dataset deduplication.

Proceedings for the Ninth Workshop on Technologies for Machine Translation of Low Resource Languages (LoResMT 2026), pages 231–234

The text cleaning stage was designed to normalize punctuation and remove artifacts that could confuse the tokenizer. This involved removing specific typographical quotation marks ("”" and "“") and removing the long dash symbol ("—") when it appeared at the beginning of a line, as it was often a residual formatting element not part of the actual sentence.

During filtering, the number of characters in the original text and its translation were counted. If the source text contained 2 times more characters, or conversely 2 times fewer characters than the translation, such a sentence was removed from the dataset. This was necessary to remove poor examples from the corpora, which could have been present, for instance, due to automatic alignment.

To ensure data quality and prevent model overfitting to repetitive content, we implemented a two-stage deduplication pipeline. First, we used the MD5 hashing algorithm from the Python standard library to efficiently identify and remove exact character-for-character duplicate sentence pairs. Subsequently, to address more subtle redundancy, we utilized the MinHashLSH algorithm from the datasketch library (Zhu et al., 2024). This probabilistic technique allowed us to detect and filter out near-duplicate examples where a significant proportion of tokens overlapped, even if the sentences were not identical. For MinHashLSH, we used a threshold of 0.7 Jaccard similarity, 128 permutations and shingle size 3.

Ultimately, this reduced the number of examples in the datasets and sped up training. The preprocessing steps removed approximately 7% of the English-Chuvash corpus. All preprocessing steps were applied consistently to all training datasets before merging, ensuring a clean and homogeneous training corpus.

3 Models

When selecting a model, we were guided by the requirement that the model should at least understand some languages from the Turkic family. The first model for the experiments was google/umt5-small (Chung et al., 2023), since the authors use their own data sampling method to prevent overfitting in data from low-resource language. Subsequently, experiments were conducted with google/gemma-270m and google/gemma-3-1b-it (Team, 2025), as it turned out the authors used a similar approach for data sampling in pretraining. Finally, experiments were conducted with the specialized translation model tencent/HY-MT1.5-7B (Zheng et al., 2025). The key characteristics of these models are summarized in Table 2.

Model	Parameters
google/umt5-small	300M
google/gemma-270m-it	270M
google/gemma-3-1b-it	1.4B
tencent/HY-MT1.5-7B	7B

Table 2: Key characteristics of the models used in our experiments.

4 Training

Various frameworks and libraries were used for model training, including Transformers (Wolf et al., 2020) to train google/umt5-small, Unsloth (Daniel Han and team, 2023) to train models from the gemma-3 family and LLaMA-Factory (Zheng et al., 2024) to train tencent/HY-MT1.5-7B. We also conducted experiments with both full fine-tuning and QLoRA fine-tuning for the larger models (gemma-3-4b-it and HY-MT1.5-7B).

For all models, we employ a standard sequence-to-sequence training objective, maximizing the likelihood of the target Chuvash translation given the English source. For encoder-decoder architectures, we prepended a simple instruction prefix (e.g., translate English to Chuvash:). For decoder-only chat models like Gemma, we formatted the input using the model's prescribed chat template, placing the translation instruction and source text within a single user message.

In experiments with the google/umt5-small and gemma-3 family models, training lasted up to 3 epochs, while in experiments with tencent/HY-MT1.5-7B, training lasted 1 epoch. A learning rate of 2e-5 with a linear scheduler was used for full fine-tuning of the google/umt5-small and gemma-3 family models, the same learning rate with a cosine scheduler was used for full fine-tuning of tencent/HY-MT1.5-7B, and a learning rate of 4e-4 with a linear scheduler was used for QLoRA configurations. The training dataset was split into train and test subsets in a 9:1 ratio. We used a batch size of 16; when a full batch did not fit in GPU memory, we employed gradient accumulation to achieve an effective batch size of 16. To reduce GPU memory requirements and speed up training, we utilized techniques such as the 8-bit AdamW

optimizer, training in pure bf16 mode, and Flash Attention 2 (Dao, 2024).

5 Generation Params

To improve the quality of the generation after fine-tuning, we selected the generation parameters on the FLORES+ dataset. For this purpose, the dataset was split equally into train and test sets. On the train set, we iterated over the generation parameters and optimized the chrF++ metric using Optuna (Akiba et al., 2019); the final score was calculated on the test set. Since the dataset is sufficiently diverse and covers a wide range of topics, the improvement in the metric on this dataset showed good correlation with the metric on the public set of the shared task.

The parameters optimized included beam search width, length penalty, repetition penalty, and temperature sampling. The optimal configuration found significantly improved translation fluency and adequacy. The best parameters for the HY-MT1.5-7B model were: temperature=0.2764731626394478, top_p=0.99559242372123, top_k=77, num_beams=5, repetition_penalty=1.0359827344136177.

This tuning process was crucial, as the default generation parameters often produced overly conservative or repetitive translations for the low-resource language. The optimized parameters encouraged more diverse and contextually appropriate outputs, which was reflected in the significant metric gain.

6 Results and Discussion

On the public leaderboard, the tencent/HY-MT1.5-7B model performed the best, and it also had the most parameters. Furthermore, tuning the generation parameters improved the quality of this version in the public set from 25.42 to 29.67 (see Table 3). This represents a relative improvement of over 16% from parameter tuning alone.

During experiments with QLoRA fine-tuning, we found that the quality of training gemma-3-4b-it was no better than that of full fine-tuning of gemma-3-1b-it, suggesting that for the scale of data available, the benefits of a larger model architecture may be offset by the limitations of the parameter-efficient fine-tuning method in this specific low-resource scenario. This observation aligns with recent findings that the effectiveness of PEFT methods can be task- and data-size dependent, and full fine-tuning may still be preferable when computational resources allow and the target data distribution differs significantly from the pretraining data.

Model	chrF++ score
umt5-small	13.03
gemma-3-270m-it	14.60
gemma-3-1b-it	20.43
gemma-3-4b-it w/ QLoRA	20.43
HY-MT1.5-7B	25.42
HY-MT1.5-7B w/ gen params	**29.67**

Table 3: Submission results on the LoResMT 2026 EnglishChuvash public test set. The best score is in bold.

The results demonstrate a clear correlation between model size and translation quality for this task. The specialized translation architecture of HY-MT1.5-7B likely contributed to its superior performance. The effectiveness of generation parameter tuning highlights the importance of inference-time optimization beyond just model training. An error analysis on a sample of outputs revealed that the larger model with tuned parameters produced fewer grammatical errors and better captured Chuvash morphology.

The parameters found in Section 5 were used for the final submission of the best model.

7 Conclusion

This paper presented our submission to the LoResMT 2026 English-Chuvash translation task. We explored various model architectures, from smaller encoder-decoder models to large multilingual translation models. Our experiments showed that the tencent/HY-MT1.5-7B model, when combined with careful data preprocessing and optimized generation parameters, achieved the highest chrF++ score of 29.67. The significant gain from the tuning of the generation parameters underscores its importance in low-resource MT pipelines. Our work also highlights the trade-offs involved in model and method selection for low-resource settings: while large, specialized models yield the best results, efficient fine-tuning techniques on similar-sized models may not provide a commensurate advantage with limited data. Future work could involve exploring additional data augmentation techniques for Chuvash, investigating more efficient

fine-tuning methods for very large models, and incorporating linguistic features specific to Turkic languages to further improve translation quality.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Hyung Won Chung, Xavier Garcia, Adam Roberts, Yi Tay, Orhan Firat, Sharan Narang, and Noah Constant. 2023. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. In *The Eleventh International Conference on Learning Representations*.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

Alexander Jones, Isaac Caswell, Orhan Firat, and Ishank Saxena. 2023. "GATITOS: Using a New Multilingual Lexicon for Low-resource Machine Translation". In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 371–405, Singapore. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Gemma Team. 2025. Gemma 3.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mao Zheng, Zheng Li, Tao Chen, Mingyang Song, and Di Wang. 2025. Hy-mt1.5 technical report. *Preprint*, arXiv:2512.24092.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Eric Zhu, Vadim Markovtsev, Aleksey Astafiev, Arham Khan, Chris Ha, Wojciech Łukasiewicz, Adam Foster, Sinusoidal36, Spandan Thakur, Stefano Ortolani, Titusz, Vojtech Letal, Zac Bentley, fpug, hguhlich, long2ice, oisincar, Ron Assa, Senad Ibraimoski, and 8 others. 2024. ekzhu/datasketch: v1.6.5.