Neural Machine Translation for Russian-Tatar Language Pair

Vsevolod Antonov Denis Salikiev

June 2025

Abstract

Neural Machine Translation (NMT) currently represents the state-of-the-art in machine translation. However, developing translation systems for low-resource language pairs remains a significant challenge due to the limited availability of parallel corpora. In this project we aim to make a Neural Machine Translator for low-resource Russian-Tatar language pair. We apply such technique as transfer learning using NLLB as the base multilingual machine translation model. This model was trained on over than 200 languages, including Turkic languages, such as Tatar. The results show that fine-tuning NLLB model for Russian-Tatar language pair using a small parallel corpus as 311,000 sentences may be not enough for solid quality improvement. Project code: https://github.com/RU-TT-translator/RU-TT-translator

1 Introduction

Making translators for low-resource languages is an important task, since it is able to make such language pairs available for global communication, create new opportunities for language learning enthusiasts and arise interest in other cultures.

There were used various approaches in area of development and enhancement of machine translation systems for low-resource languages, such as Tatar, including the use of multilingual pre-trained models and transfer learning, various methods of augmentation and quality impovement of training data by pseudo-labeling and back-translation. This task included preparing training and test datasets, training and evaluation of the model.

1.1 Team

Data preparation, model training and report writing were done by Vsevolod Antonov.

Related work research, creation of pipeline and code for model training, its comparison with related models and report writing were done by Denis Salikiev.

2 Related Work

A. Research at Innopolis University [1]

In this study various advanced techniques for machine translation improvement for low-resource languages on the example of the Russian-Tatar pair were discussed. The main methods described in this paper are:

- Transfer Learning: This approach involves initial training in a comparably high-resource language pair (for example, Russian-Kazakh), and then fine-tuning it to Russian-Tatar. Such method helps to transfer knowledge from a high-resource task to a low-resource one, improving performance. After the NLLB-200 model appears, we can successfully transfer new knowledge to it, which, presumably, will give better quality.
- Back-translation: This semi-peer training method involves translating monolingual target sentences into the source language to create synthetic parallel data. This synthetic data is then used together with the original parallel corpus to train the NMT model. Back-translation lets to improve the language model on the target language side, improving the fluency and coherence of translated text.
- Language Model Integration (Shallow Fusion): This method includes a separately trained language model in the NMT model during decoding. The language model trained on monolingual data provides additional context, improving the fluency and grammaticality of the output.

The authors show that combining these techniques leads to significant improvement in standard translation metrics for both Russian-to-Tatar and Tatar-to-Russian translations. In particular, the use of transfer learning and back-translation showed noticeable improvements, while the language model integration through shallow fusion made less impact.

B. Research at Institute of Applied Semiotics Tatarstan Academy of Sciences [2]

In this paper scientists relied on previous discussed research fine-tuned pre-trained model NLLB-200. They used the following techniques for training:

Mixed Precision Training (MPT) FP16 They applied a neural network training method that uses mixed precision to speed up calculations and reduce memory usage. [3] In this method, as they say calculations are performed using the 16-bit floating-point format (FP16).

Efficient Sharding

A method that is used to reduce memory consumption when training large neural networks. The basic idea is to separate model weights and activations (intermediate values) across multiple devices, such as graphics processing units (GPUs). This allowed them to efficiently use available resources and process models that would otherwise not fit in the memory of a single device.

Pseudo-labeling

This is a technique that is used to improve the quality of machine learning models by using unlabeled data. For example, in the context of machine translation, the translation model is first trained on parallel data (i.e. pairs of sentences in the source and target languages), and then this model is used to translate sentences from an unmarked monocorpus into the language we need. Studies on this topic [4] show that such an approach can really give an increase in quality.

Back-translation

In addition to pseudo-labeling, authors also applied the back-translation method to increase the data volume [5]. This is one of the most widely used approaches in the field of natural language processing, especially in the tasks of machine translation and generating additional data. The method consists in translating the text into another language and then returning it to the original language, thus creating additional training data.

Beam Search

Beam search [6] is a method of text generation, which algorithm is a modification of greedy search and is designed to find the optimal sequence of solutions in problems where a choice is needed at each stage. The main idea of beam search is to maintain a fixed number of best candidates (beam width) at each stage of the sequence construction. Unlike greedy search, which continues with only the best immediate branch, beam search tracks multiple probable paths simultaneously, increasing the likelihood of finding a globally optimal solution

3 Model Description

Our approach incorporated three techniques which were discussed in previous works:

- **Fine-tuning the NLLB-200 distilled 600M model** [7], which was pre-trained on over 200 languages, including Tatar.
- **Mixed precision training (MPT) FP16** to provide speeding up calculations and reducing memory usage.
- **Beam-search at inference time** to improve model quality at inference time. As was mentioned earlier, it tracks multiple probable paths simultaneously, increasing the likelihood of finding a globally optimal solution

Below NLLB-200 distilled 600M architecture, the data it was trained on, and its main features are described:

1. Transformer Backbone

NLLB-200 is based on the Transformer architecture [8], which stacks encoder and decoder layers and uses attention mechanisms for sequential data processing.

2. Mixture-of-Experts (MoE)

A defining feature of NLLB-200 is its integration of Mixture-of-Experts (MoE) blocks [9]. This technique incorporates specialized submodels ("experts") within the architecture, where only relevant experts activate per input. This selective activation maintains model scalability while optimizing computational efficiency.

3. Encoder-Decoder Workflow

The model follows a standard encoder-decoder design with loss function equal to cross-entropy loss + MoE:

- The encoder transforms source text into contextual representations.
- The decoder generates target-language output from these representations.

Both employ multi-layer self-attention to capture full-sentence context during translation.

4. Multilingual Capability
NLLB-200 translates across 200+ languages within a unified framework. Target languages are specified using unique identifier tokens, enabling the model to dynamically route inputs to language-specific processing pathways.

5. SentencePiece Tokenization
Text is tokenized via SentencePiece, which supports two subword segmentation algorithms:

- Byte-Pair Encoding (BPE) [10]: Merges frequent character pairs iteratively.
- Unigram Language Model: Uses probability to determine optimal splits.

Unlike space-dependent tokenizers, SentencePiece processes text as a continuous character stream.

Beam width was set to 8 based on findings in [2], which showed it offered the best trade-off between quality and inference time.

4 Dataset

We compiled a Russian-Tatar parallel corpus from multiple sources, initially comprising 402,188 sentence pairs. After preprocessing, the dataset was reduced to 311,289 pairs. The test set includes 2,009 sentence pairs from FLORES-200 dataset [7].

Used datasets:

1. AigizK/tatar-russian-parallel-corpora — parallel corpus developed by Academy of Sciences of the Republic of Tatarstan, Institute of Applied Semiotics.
2. OpenSubtitles - collection of translated movie subtitles.
3. Wikimedia — Wikipedia translations published by the wikimedia foundation and their article translation system.
4. Tanzil — a collection of Quran translations compiled by the Tanzil project.
5. Tatoeba — a free collection of example sentences with translations geared towards foreign language learners.

We used the following algorithm for data processing;

1. Removal of all rows containing symbols from the Latin alphabet.
2. Removal of all text inside of and including parentheses, HTML, e-mail, URL, emoji, special symbols, multiple and trailing spaces.
3. Conversion of all text to lowercase.
4. Removal of None values, duplicates and empty rows.
5. Filtering out sentences containing more than 25 words.
6. Removal of all rows with tatar text containing less than 2 specific tatar letters in order to filter it by language.

Table 1. Dataset size

Rows number before processing	402188
Rows number after processing	311289

Table 2. Dataset split

train	307109
val	4180
test	2009

5 Experiments

5.1 Metrics

The metrics used for the models evaluation are the SacreBLEU and ChrF++. SacreBLEU's core formula is the same as for the classic BLEU:

$$BLEU = \min(1, \frac{length\ of\ MT}{length\ of\ reference}) \times \left(\prod_{i=1}^{n} Precision_i\right)^{\frac{1}{n}}$$

$$Precsision_i = \frac{clipped\_number\_of\_matched\_\{i\}grams\_in\_MT}{number\_of\_total\_\{i\}grams\_in\_MT}$$

The first factor is brevity penalty. It's used to avoid a very short machine translation getting a high BLEU score. The second factor is N-gram precision.
The number of matched $\{i\}grams$ is clipped to avoid that a single word in the reference is matched multiple times in MT.

The difference is that BLEU is tokenization-dependent. SacreBLEU (Post, 2018) [11] have taken steps towards standardization, supporting utilizing community-standard tokenizers under the hood.

ChrF++ uses the F-score statistic for character n-gram matches and adds word n-grams which correlates more strongly with direct assessment.

The general formula for the CHRF score is:

$$CHRF\beta = \frac{(1 + \beta^2)(CHRP \cdot CHRR)}{(\beta^2 \cdot CHRP + CHRR)}$$

where CHRP and CHRR stand for character n-gram precision and recall arithmetically averaged over all n-grams:

- CHRP is percentage of n-grams in the hypothesis which have a counterpart in the reference;
- CHRR is percentage of character n-grams in the reference which are also present in the hypothesis;

and β is a parameter which assigns β times more importance to recall than to precision – if β = 1, they have the same importance.

5.2 Experiment Setup

Hyperparameters of learning:
Training was performed on Kaggle using a P100 GPU. To learn faster we fine-tuned separate models for each translation direction: Russian → Tatar and Tatar → Russian. Both models were trained for 2 epochs. Initial hyperparameters (used in both the full training of Tatar → Russian and the first epoch of Russian → Tatar) included:

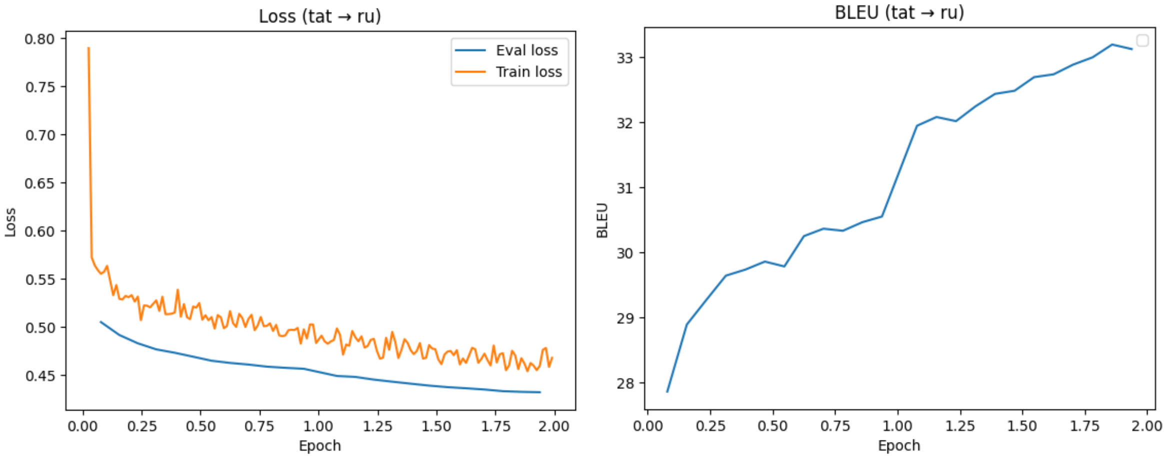
batch_size: 8 — batch size
num_train_epochs: 2 — number of training epochs
weight_decay: 1e-2 — regularization coefficient of AdamW
learning_rate: 2e-5 — learning rate
adam_beta1: 0.9 — parameter of the optimizer
adam_beta2: 0.999 — additional parameter of the optimizer
adam_epsilon: 1e-8 — parameter to prevent division by zero
num_beams: 8 — the number of "rays” in the beam search algorithm when making predictions in inference
max_source_length: 256 — maximum length of a sentence in the source language in tokens
max_target_length: 256 — maximum sentence length in the desired language in tokens
fp16: True — parameter for mixed precision training

Parameters for model for Russian → Tatar translation direction for the 2nd epoch training were changed because loss value ceased to decrease towards the end of the 1st epoch:

learning_rate: 8e-6 — decreased learning rate
weight_decay: 5e-3 — decreased regularization coefficient of AdamW
lr_scheduler_type: 'cosine' — scheduler of learning rate
warmup_ratio: 0.05 — defines the proportion of the training process that is dedicated to the warmup phase.

6 Results

In this section, we show graphics of the training process and a comparison of our model's quality against the NLLB200 600M base model (before fine-tuning) and Yandex Translator.



Figire 1. Loss and quality of the TT-RU translation model in the course of training

As one can see on the Figure 1, while both losses for TT-RU model decrease steadily, the training loss starts higher and exhibits more variability, especially early on. The evaluation loss consistently decreases in a smoother fashion. This suggests that the model is learning, but the slower and noisier convergence on the training set may indicate an imbalance in training data quality.

The BLEU score rises consistently from 28 to over 33 across 2 epochs, showing consistent improvement in translation quality without signs of overfitting. The upward trend suggests the model is learning effectively and could benefit from further training.

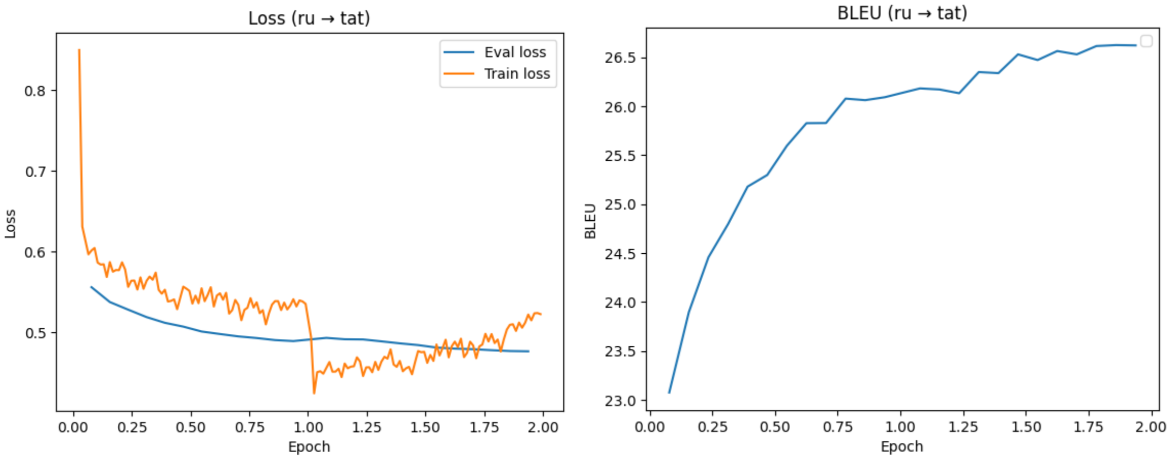


Figure 2. Loss and quality of the RU-TT translation model in the course of training

Figure 2 shows that the training loss for RU-TT model initially decreases rapidly but begins to plateau toward the end of the first epoch. In response, hyperparameters were adjusted after epoch 1, resulting in a sharp drop in training loss. After the change, the loss becomes more variable, while evaluation loss continues to decline steadily, suggesting overall progress but with some training instability.

BLEU scores improve steadily from 23 to about 26.5, with the sharpest gains occurring in the first epoch. Progress continues after the hyperparameter adjustment but at a slower, more incremental pace, suggesting diminishing returns but continued learning.

Table 3. Evaluation of model

Translation direction	Yandex		NLLB-200. 600 before fine-tuning		NLLB-200. 600 after fine-tuning (our)	
	chrF++	sacreBLEU	chrF++	sacreBLEU	chrF++	sacreBLEU
RU-TT	35.212	6.45	**45.004**	15.303	44.175	**16.652**
TT-RU	34.561	7.375	**45.571**	**20.511**	43.355	18.964

Evaluation on the FLORES-2000 benchmark revealed that, for the Russian→Tatar (RU→TT) direction, our approach yielded a sacreBLEU improvement of +1.349 points relative to the baseline, whereas the chrF++ score declined by −0.829 points. In the reverse (Tatar→Russian) direction, both sacreBLEU and chrF++ metrics indicated a net degradation in translation quality compared to the baseline system. Although precise interpretation of these mixed results remains elusive, we hypothesize that a substantial mismatch between the training and evaluation corpora—particularly regarding sentence-length distributions and contextual complexity—has hindered consistent gains across both metrics and language pairs. Specifically, the FLORES-2000 test set comprises a higher proportion of longer, syntactically complex sentences than our cleaned training data, which was heavily filtered to exclude sentences over 25 words and non-Tatar content. Consequently, while the model may have learned to translate shorter, more homogeneous sentences effectively (reflected in the sacreBLEU gain for RU→TT), it may have underperformed on longer, multi-clausal structures, leading to the observed drop in chrF++.

7 Conclusion

In this study, we assembled and preprocessed a parallel Russian–Tatar corpus, and leveraged it to fine-tune a modified NLLB-200 600M model. Our preprocessing pipeline encompassed the following specific cleaning steps to ensure high-quality training data: removal of all rows containing symbols from the Latin alphabet; removal of any text inside and including parentheses, as well as HTML tags, e-mail addresses, URLs, emojis, special symbols, and multiple or trailing spaces; conversion of all text to lowercase; removal of “None” values, exact duplicates, and empty rows; removal of any sentence pair in which either side contained more than 25 words; and removal of all rows whose Tatar segment contained fewer than two language-specific Tatar letters, in order to filter out non-Tatar content. During model training, we incorporated mixed-precision optimization to accelerate convergence and enable larger batch sizes on available hardware. At inference time, we employed beam-search decoding to improve translation quality.

Our evaluation on the FLORES-2000 benchmark showed mixed results: while the RU→TT direction saw a modest sacreBLEU improvement, it also experienced a decline in chrF++, and the TT→RU direction showed overall performance degradation. These inconsistencies likely stem from a mismatch between the training data—filtered for short, simple sentences—and the more complex evaluation set, suggesting the need for more balanced and representative training corpora to ensure consistent translation quality.

Looking ahead, we intend to undertake several enhancements aimed at reducing domain-shift and improving overall translation fidelity. First, we will expand our parallel corpus by sourcing larger volumes of aligned Russian–Tatar data from diverse domains. Second, we plan to extend training schedules by increasing the number of epochs and experimenting with adaptive learning-rate schedules, thereby allowing the model additional opportunities to converge on hard-to-translate patterns. Third, we will revisit our case-normalization strategy: although this study uniformly converted all text to lowercase, future experiments will explore preserving case information to capture named entities and morphological cues vital in both Russian and Tatar.

References

1. A. Valeev, I. Gibadullin, A. Khusainova, and A. Khan, "Application of Low-resource Machine Translation Techniques to Russian-Tatar Language Pair", arXiv preprint arXiv:1910.00368, 2019. [Online]. Available: https://arxiv.org/pdf/1910.00368
2. R. Gilmullin, B. Khakimov, R. Leontiev and S. Trifonov. (2024). Development of the Russian-Tatar Translation Model Based on NLLB-200. 1900-1905. 10.1109/PIERE62470.2024.10804955.
3. S. Narang, G. Diamos, E. Elsen, P. Micikevicius, J. Alben, D. Garcia, et al, "Mixed Precision Training," in Proceedings of the International Conference on Learning Representations (ICLR), vol. 2, 2018, pp. 1086-1097. [Online]. Available: https://arxiv.org/pdf/1710.03740
4. B. Hsu, A. Currey, X. Niu, M. Nădejde, G. Dinu, " Pseudo-Label Training and Model Inertia in Neural Machine Translation," in Proc. 11th International Conference on Learning Representations, 2023. [Online]. Available: https://arxiv.org/abs/2305.11808
5. S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding Back-Translation at Scale," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2018, pp. 489–500. [Online]. Available: https://aclanthology.org/D18-1045.pdf.
6. M. Freitag and Y. Al-Onaizan, "Beam Search Strategies for Neural Machine Translation," in Proceedings of the Workshop on Neural Machine Translation and Generation, 2017, pp. 30–39. [Online]. Available: https://aclanthology.org/W17-3207.pdf.
7. NLLB Team, "No Language Left Behind: Scaling Human-Centered Machine Translation," *arXiv preprint arXiv:2207.04672*, 2022. [Online]. Available: https://arxiv.org/abs/2207.04672.
8. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems 30 (NIPS 2017), Von Luxburg, U. et al. Eds. New York, USA, 2017, pp. 5999-6009. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/file/3f5ee243547de e91fbd053c1c4a845aa-Paper.pdf.
9. N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le et al., "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," in Proc. 5th International Conference on Learning Representations (ICLR 2017), vol. 2. New York, USA, 2017, pp. 878-896. [Online]. Available: https://arxiv.org/pdf/1701.06538.

10. R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in Proc. 54th Annual Meeting of the Association for Computational Linguistics, vol. 1. Berlin, Germany, 2016, pp. 1715-1725. [Online]. Available: https://aclanthology.org/P16-1162.pdf
11. Post, Matt. (2018). A Call for Clarity in Reporting BLEU Scores. 10.48550/arXiv.1804.08771.