DOI: 10.14529/jsfi250401

Distillation for Adaptation Language Models to Russian Language

Grigory P. Kovalev[1], *Mikhail M. Tikhomirov*[1]

Adapting large language models (LLMs) to morphologically rich languages like Russian presents a major challenge, as multilingual models often exhibit limited transfer due to predominantly English-centric pre-training. This study investigates knowledge distillation (KD) as a more effective alternative to supervised fine-tuning (SFT) for the final calibration stage of language adaptation. We introduce an efficient offline top-K distillation approach that transfers knowledge from a 32B Russian-adapted teacher model to a 4B student model through tokenizer alignment and direct logit transfer. Experimental results demonstrate that KD consistently surpasses SFT, achieving up to a 4.22% performance improvement, with top-100 distillation yielding the highest gains (3.27% on average) albeit with increased memory consumption (62 GB vs. 7 GB for top-10). Moreover, the advantages of KD are most pronounced for student models with lower adaptive capacity (i.e., smaller LoRA α values). These findings underscore the efficacy of KD as a practical and scalable approach for language adaptation, while emphasizing the necessity of balancing performance improvements against computational efficiency.

Keywords: large language model, distillation, fine-tuning, model adaptation, Russian language.

Introduction

Large Language Models (LLMs) such as Qwen [20], DeepSeek [3], Llama [4], and GPT [1] have rapidly advanced the state of the art in NLP. Despite nominal multilinguality, their pre-training data is heavily dominated by English, which limits performance in underrepresented languages. The gap is particularly pronounced for morphologically rich languages like Russian, where inflectional complexity and orthographic variation amplify subword fragmentation under default tokenizers. As a result, language adaptation – porting an existing LLM to a target language and tokenizer while preserving its instruction alignment and skills – remains a practical necessity.

A common adaptation pipeline involves extending the model's tokenizer, performing continued pre-training, and then conducting supervised fine-tuning (SFT) to align the model with language-specific instructions. While this approach improves fluency, SFT relies on hard targets and may not be the most effective way to transfer knowledge, especially when adapting a smaller model under the guidance of a much larger, more capable one. A more potent alternative is knowledge distillation, where a "student" model learns from the full output distribution of a "teacher" model. However, distillation is often complicated by tokenizer mismatches, making direct logit transfer between different models problematic.

This paper investigates logit distillation as a superior alternative to SFT within a language adaptation pipeline. Our core methodology resolves the tokenizer mismatch problem by first aligning the vocabularies of the teacher and student models through a shared extension of Russian-specific tokens. This enables a direct and clean transfer of knowledge. Concretely, we leverage a pre-existing 32B Qwen3[2] model, which was fully adapted for Russian. We then use it as a teacher to guide the adaptation of a 4B Qwen3[3] student model. Our primary contribution

[1] Lomonosov Moscow State University, Moscow, Russian Federation

[2] `https://huggingface.co/RefalMachine/RuadaptQwen3-32B-Instruct`

[3] `https://huggingface.co/Qwen/Qwen3-4B`

is a comprehensive comparison showing that replacing the final SFT stage with tokenization-aligned logit distillation results in a more powerful and efficient Russian language model.

To ensure reproducibility and to facilitate the development of language-specific models, we publicly release our code on GitHub[4].

The remainder of this paper is organized as follows. Section 1 reviews prior work on multilingual large language model adaptation and knowledge distillation. Section 2 describes the model adaptation pipeline based on tokenization alignment and the Learned Embedding Propagation (LEP) technique. Section 3 presents the proposed logit distillation approach designed to improve training efficiency and stability. Section 4 details the datasets used for adaptation and fine-tuning, while Section 5 outlines the evaluation framework and benchmarks employed to assess model performance across diverse linguistic and reasoning tasks. Section 6 reports experimental results and analysis. Section 7 discusses the main limitations of our study, and the paper concludes with a summary of key findings and outlines directions for future research.

1. Background

Several adaptation methods have been proposed to overcome the limitations of multilingual LLMs for languages such as Russian. The most straightforward method for such adaptation is supervised fine-tuning (SFT) on target-language instructions [16]. A more complex but efficient variant is to perform tokenization vocabulary adaptation [21] before the SFT step, which can better align the model's internal representations with the target language's morphology. The fine-tuning process itself is an active area of research, with methods including classical SFT, reinforcement learning from human feedback (RLHF) to align outputs with human preferences [17], and knowledge distillation, where a smaller "student" model learns from a larger "teacher" models outputs to transfer knowledge efficiently [10]. Among these, knowledge distillation is particularly promising for language adaptation, as it enables the creation of compact, language-specific models without full retraining. However, fine-tuning is not without challenges, as modern LLMs have dense knowledge distributions, and improper fine-tuning can lead to forgetting, where the model loses previously learned capabilities.

2. Model Adaptation

Our work builds upon the methodology proposed by Tikhomirov et al. [21] for adapting large language models (LLMs) to target languages, with a focus on addressing the challenges posed by morphologically rich languages like Russian. This approach systematically modifies the model's tokenization and internal representations to better capture language-specific nuances, followed by a calibration phase to optimize performance on target-language tasks. The adaptation process consists of the following steps:

1. **Construction of a new tokenization vocabulary:** A language-specific vocabulary is created to account for the morphological and linguistic characteristics of the target language, augmenting the original tokenizer's vocabulary.
2. **Training embeddings for new vocabulary elements:** New token embeddings are trained to represent the added vocabulary items, ensuring compatibility with the model's architecture and preserving semantic richness.

[4] `https://github.com/RefalMachine/ruadapt`

3. **Alignment of the base language model with the new vocabulary:** The model's parameters are adjusted to align with the updated tokenizer, ensuring seamless integration of the new embeddings into the model's existing knowledge [21].
4. **Transfer of core linguistic knowledge:** Core linguistic knowledge is transferred to the target version of the model using the Learned Embedding Propagation (LEP) method [21].
5. **Calibration of the adapted model:** The adapted model is fine-tuned on task-specific examples in the target language to optimize performance.

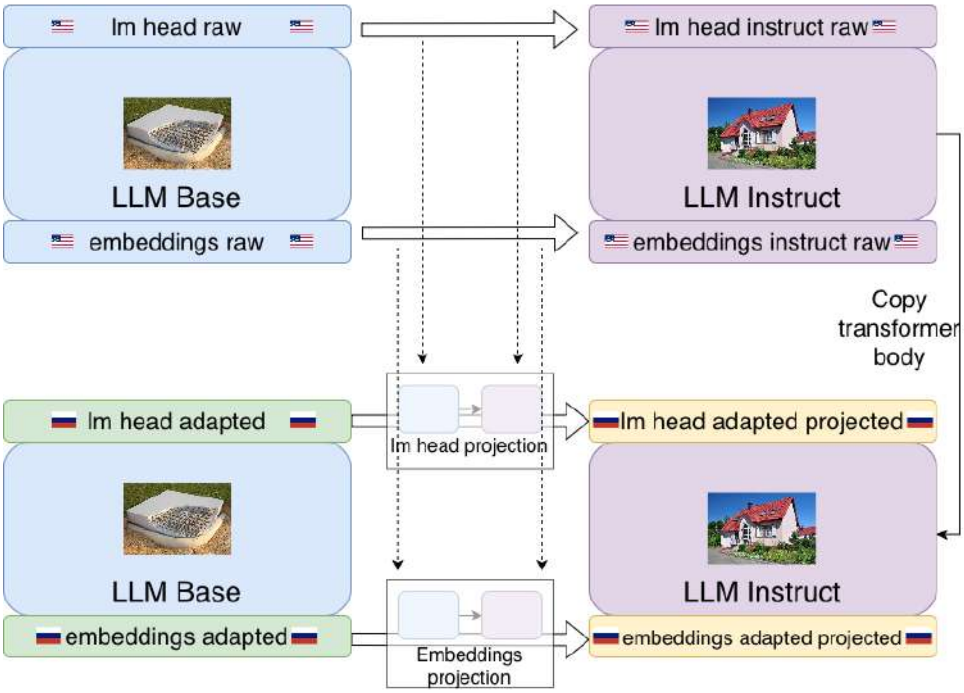

Figure 1. Language adaptation scheme

The fifth step – calibration of the adapted model – is the primary focus of our research. Calibration is critical, as it ensures that the model not only understands the target language's linguistic structures but also performs effectively on downstream tasks. To address this, we explore efficient yet computationally effective calibration methods, comparing classical supervised fine-tuning (SFT) with knowledge distillation. Classical SFT directly optimizes the model on labeled target-language data, while knowledge distillation transfers task-specific knowledge from a larger teacher model to a smaller student model.

In our experiments, we selected a relatively small student model to enable extensive experimentation, thereby providing deeper insights into the benefits of our proposed methodology. Specifically, we used an adapted version of the 4B Qwen3 model, evaluating its performance on both English and Russian-specific benchmarks to assess the trade-offs between efficiency, computational cost, and task performance. Our findings aim to provide practical guidelines for adapting LLMs to low-resource languages with complex morphologies.

3. Distillation Methodology

Knowledge distillation is a well-established technique for transferring knowledge from a larger "teacher" model to a smaller "student" model, enabling efficient model compression while preserving performance [8, 10, 11, 15]. Its flexibility lies not only in the variety of distillation methods but also in the ability to experiment with the teacher model's configuration and outputs.

As previously mentioned, the student model is an adapted version of the 4B Qwen3 following the Learned Embedding Propagation (LEP) step [21]. In this step, we applied the LEP procedure described in [21] to propagate the newly learned token embeddings and linguistic knowledge from the base 4B Qwen3 model to the instruction-tuned 4B variant. The method approximates full re-training with a newly adapted tokenizer through a series of learned linear operators that align embedding spaces between the source (multilingual) and target (Russian-adapted) vocabularies. Before LEP, the base model underwent re-tokenization with an extended Unigram-based vocabulary optimized for Russian morphology and subsequent fine-tuning of embeddings and internal layers on approximately 150 GB of Russian text data constructed as a combination of the rulm [9] and Fineweb-2 [18] corpora. LEP then transferred these updated embeddings and LoRA adapter weights onto the instruction-tuned 4B checkpoint, allowing the resulting model to preserve instruction-following capabilities while achieving native-level Russian tokenization and representation quality. This approach substantially reduces the computational cost of full re-training.

For the "teacher" model, we chose RuadaptQwen3-32B[5], which has been pre-adapted for Russian-language tasks and is currently the largest and most capable publicly available model using the same tokenization.

Simultaneously storing both models in memory for distillation is challenging due to their sizes, and directly training the student model on the teacher's full output distributions is computationally expensive, especially when aiming for an efficient adaptation methodology that balances quality and resource demands.

To address these challenges, we adopt a top-k offline distillation approach [2, 19], which separates the teacher logit generation and student training phases to reduce computational overhead. In the first step, we generate and store the top-k logits produced by the teacher model for each token in the training dataset, where k is a tunable hyperparameter that controls the trade-off between information retention and memory efficiency. This precomputation eliminates the need to load the teacher model during student model training, significantly reducing memory requirements. In the second step, we train the student model using these precomputed logits. To further enhance efficiency, we employ parameter-efficient fine-tuning via Low-Rank Adaptation (LoRA) adapters, which minimize the number of trainable parameters while maintaining performance [12]. The student model is trained on the same dataset used to generate the teacher's logits, ensuring consistency in the knowledge transfer process.

Our training objective combines two loss functions: (1) a classical supervised fine-tuning (SFT) loss, specifically Cross-Entropy between the true tokens and the student's predicted tokens, to calibrate the model for Russian-specific tasks; and (2) a Kullback–Leibler Divergence (KLDivLoss) term that aligns the student's output distribution with the teacher's precomputed top-k logits [10]. This combination enables the student model to learn both from ground-truth data and the teacher's "dark knowledge" – patterns in the teacher's output probabilities that enhance generalization [10]. Our combined loss function adopts the formulation proposed by Raman et al. [19], integrating Cross-Entropy and Kullback–Leibler Divergence to balance task-

[5] `https://huggingface.co/RefalMachine/RuadaptQwen3-32B-Instruct`

specific calibration and knowledge transfer from the teacher model.

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{CE}} + \lambda \left(1 - \exp\left(-\frac{s_i}{t_i + \varepsilon}\right)\right) \mathcal{L}_{\text{KD}}, \tag{1}$$

where:

- $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss between the student predictions and the ground-truth labels;
- $\mathcal{L}_{\text{KD}}$ is the knowledge distillation loss, computed as the KL divergence between the students and teachers probability distributions over the top-K tokens;
- s_i is the student logit corresponding to the ground-truth token at position i;
- t_i is the teacher logit for the ground-truth token at position i (if the token is not in the teachers top-K, t_i is set to 0);
- ε is a small constant added for numerical stability;
- λ is a scaling coefficient (denoted as `loss_multi` in our implementation).

4. Data

The quality of training data is paramount in any training process, whether pre-training or fine-tuning, as it directly influences model performance. Fine-tuning, in particular, is a delicate process, as inconsistencies or contradictions between new training data and the data used for prior training can result in neutral or even negative outcomes. To ensure effective fine-tuning for Russian language adaptation, we selected the `RefalMachine/ruadapt_hybrid_instruct` dataset[6], which comprises approximately 80,000 instruction samples tailored for Russian-language tasks.

This dataset was originally introduced as part of the RuAdapt framework[7] and was specifically designed to serve as calibration data for Russian instruction-tuned models. It consists of high-quality synthetic examples generated by the `Qwen3-235B-A22B` model using prompts drawn from the GrandMaster-PRO-MAX [16] collection. For each prompt, three candidate responses were produced, and the shortest valid response written in Russian and free of non-cyrillic symbols was retained. The resulting dataset captures a broad spectrum of instruction-following behaviors while maintaining linguistic consistency with the target language.

5. Evaluation

For evaluation in our experiments, we utilized the `llmtf` framework[8], an open-source toolkit designed to assess the performance of instruction-tuned language models in both few-shot and zero-shot scenarios. This framework supports flexible evaluation across diverse tasks, enabling comprehensive analysis of model capabilities on Russian-specific benchmarks. The `llmtf` framework also standardizes prompt templates and scoring procedures, ensuring reproducibility of the reported results.

To obtain a balanced picture of model performance, we evaluated the models in the zero-shot setting on a diverse suite of datasets that cover multiple linguistic and reasoning abilities. Specifically, the following benchmarks were used:

[6] https://huggingface.co/datasets/RefalMachine/ruadapt_hybrid_instruct

[7] https://github.com/RefalMachine/ruadapt

[8] https://github.com/RefalMachine/llmtf_open

- **NEREL** [13], a Russian named entity recognition benchmark derived from news and Wikipedia texts, testing the models ability to identify and classify named entities in context.
- **Summ**[9], a summarization dataset based on Russian news articles, assessing the models capacity for text compression and important information extraction.
- **MultiQ** and **USE** (from MERA [5]), multi-domain question-answering and semantic understanding benchmarks that evaluate reasoning and general comprehension abilities across diverse Russian topics.
- **Copy**, a diagnostic test of generation robustness that measures the models tendency to produce repetitive or degenerate outputs under constrained input prompts.
- **FLORES** (ru–en and en–ru) [6], a multilingual machine translation benchmark used to measure cross-lingual generalization and translation consistency between Russian and English.
- **enMMLU** and **ruMMLU**, multilingual general knowledge and reasoning benchmarks adapted from the Massive Multitask Language Understanding dataset, providing a standardized measure of factual recall and reasoning accuracy across academic domains.
- **IFEval** (en and ru versions) [22], a meta-evaluation suite for instruction-following behavior that quantifies how well a model adheres to explicit task instructions and constraints.
- **ruOpinionNE** [14], a sentiment analysis dataset focusing on Russian social media and news, testing contextual polarity and stance detection.
- **ruParam** [7], a benchmark for paraphrase and semantic similarity detection in Russian, designed to measure semantic coherence and lexical flexibility.

Together, these benchmarks cover a wide range of linguistic competencies, including named entity recognition, summarization, question answering, translation, reasoning, and adherence to instructions. This diversity allows us to evaluate both general language understanding and the effectiveness of Russian-specific adaptation. All evaluations were conducted in the zero-shot setting, using the default templates provided by `llmtf`, to ensure fair comparison across models and reproducibility of results.

6. Experiments

The distillation process is undoubtedly more computationally demanding. To assess its effectiveness, we conducted a series of experiments comparing classical supervised fine-tuning (SFT) with knowledge distillation during the calibration stage of model adaptation. Our central question is whether knowledge distillation can improve model performance over classical SFT, and whether the potential gains justify its additional cost.

6.1. Supervised Fine-Tuning

For the supervised fine-tuning (SFT) stage, we employed LoRA adapters [12]. Building on prior experiments, we fixed the LoRA rank at 128 and treated LoRA α together with the learning rate as tunable hyperparameters. Since our distillation approach introduces two additional hyperparameters – top-K and λ – we first identified the optimal values of LoRA α and the learning rate using classical SFT, and only then extended the setup to knowledge distillation.

[9] `https://huggingface.co/datasets/IlyaGusev/gazeta`

Table 1. Comparison of different hyperparameter configurations
(a – LoRA alpha, lr – learning rate) across multiple evaluation benchmarks

Config		**mean**	**NEREL**	**Summ**	**MultiQ**	**USE**	**flores**	**copy**	**ru babllong**	**en IFEval**	**ru IFEval**	**en MMLU**	**ru MMLU**	**ru opinionqa**	**ru param**
a	lr														
64	1e-4	**0.484**	**0.489**	**0.225**	**0.200**	0.034	**0.507**	0.940	0.504	**0.712**	**0.636**	0.705	0.621	0.088	**0.632**
	2e-5	0.468	0.478	0.154	0.186	0.029	0.504	**0.985**	0.490	0.688	0.560	0.706	0.620	0.088	0.600
	5e-5	0.483	0.475	0.223	0.195	**0.041**	**0.507**	0.950	**0.509**	0.704	0.621	**0.708**	**0.624**	**0.097**	0.631
128	1e-4	**0.483**	**0.495**	**0.224**	**0.194**	**0.037**	**0.508**	0.945	0.504	**0.717**	0.601	**0.706**	0.620	**0.092**	0.634
	2e-5	0.472	0.480	0.181	0.183	0.028	0.506	**0.980**	0.494	0.684	0.582	**0.706**	**0.622**	0.084	0.603
	5e-5	0.482	0.478	0.223	0.193	0.031	**0.508**	0.955	**0.507**	0.693	**0.608**	0.705	0.620	0.091	**0.653**
256	1e-4	**0.490**	**0.497**	**0.223**	**0.200**	**0.043**	**0.508**	0.940	0.510	**0.726**	**0.636**	**0.707**	**0.623**	**0.106**	**0.647**
	2e-5	0.479	0.477	0.205	0.185	0.025	0.507	**0.980**	0.498	0.702	0.584	0.706	0.621	0.100	0.634
	5e-5	0.477	0.479	**0.223**	0.195	0.031	0.507	0.955	**0.514**	0.682	0.590	0.705	0.618	0.095	0.609
512	1e-4	**0.494**	**0.503**	0.222	**0.195**	**0.039**	**0.507**	0.950	**0.516**	**0.738**	**0.617**	0.704	**0.623**	**0.112**	0.692
	2e-5	0.475	0.475	0.218	0.189	0.028	0.506	**0.970**	0.509	0.710	0.584	**0.707**	0.620	0.088	0.568
	5e-5	0.488	0.484	**0.224**	0.191	0.036	0.506	**0.970**	0.500	0.713	0.608	0.705	0.621	0.096	**0.694**

From a performance perspective, we selected the four most promising configurations for further investigation of the distillation approach. These configurations were chosen based on their average performance across a diverse set of benchmarks, ensuring robust generalization. The selected configurations are: (1) $a = 128$, lr=1$e - 4$, (2) $a = 256$, lr=1$e - 4$, (3) $a = 512$, lr=1$e - 4$, and (4) $a = 512$, lr=5$e - 5$.

6.2. Knowledge Distillation

We adopted the LoRA rank, α, and learning rate values from the selected SFT configurations, while treating top-K and λ as the main hyperparameters of interest in this series of experiments. Each chosen SFT setup was compared against eight distillation variants, with $\lambda \in \{0.1, 0.2, 0.5, 1\}$ and top-$K \in \{10, 100\}$. The top-K parameter controls the number of top-ranked teacher model predictions used in the distillation process, while λ balances the contribution of the distillation loss against the supervised loss. To ensure a fair comparison, we maintained identical training conditions (e.g., batch size, training epochs, and dataset) across SFT and distillation experiments.

The results, presented in Tables 2 to 5 (Table 6 shows more detailed results), demonstrate that knowledge distillation consistently enhances model performance over the SFT baselines. The most significant improvement was observed for the SFT baseline with $\alpha = 128$ and lr=1e-4. When distilled with top-K=100 and $\lambda = 0.5$, this model achieved an aggregate score of 0.503, a 4.22% increase over its SFT counterpart. This highlights the potential of distillation to further refine already fine-tuned models.

The Influence of top-K. A clear pattern emerges when comparing top-K values: using a larger context from the teacher model (top-100) generally yields superior results compared to a smaller one (top-10). For instance, with the $\alpha = 128$ baseline, the average improvement for top-100 variants was 3.27%, compared to 2.82% for top-10. However, this performance gain comes at a significant computational cost. Storing precomputed logits for top-100 required 62 GB of memory, whereas top-10 required only 7 GB – an 8.9-fold reduction. This trade-off makes top-10 a resource-efficient option for achieving modest gains, while top-100 is preferable when maximizing performance is the priority and resources permit.

Tuning the Distillation Weight λ. Our experiments show that $\lambda = 0.2$ emerges as a robust and generally effective choice for the distillation weight. It delivered the highest performance gains in the majority of our tested configurations. However, the single best result (4.22% growth) was achieved with $\lambda = 0.5$ in the $\alpha = 128$ setup. This suggests that while $\lambda = 0.2$ is a strong starting point for tuning, the optimal value can vary depending on other hyperparameters.

Table 2. SFT a=128, lr=1e-4

Base Config	Variant	Aggregate Score	Growth, %
SFT a=128, lr=1e-4	—	0.483	—
a=128, lr=1e-4, top-10	λ=0.1	0.498	3.17
	λ=0.2	0.497	2.95
	λ=0.5	0.495	2.56
	λ=1	0.495	2.58
Avg. Growth Top-10			**2.82**
Max. Growth Top-10			**3.17**
a=128, lr=1e-4, top-100	λ=0.1	0.498	3.20
	λ=0.2	0.498	3.15
	λ=0.5	***0.503***	***4.22***
	λ=1	0.495	2.50
Avg. Growth Top-100			**3.27**
Max. Growth Top-100			**4.22**

Table 3. SFT a=256, lr=1e-4

Base Config	Variant	Aggregate Score	Growth, %
SFT a=256, lr=1e-4	—	0.490	—
a=256, lr=1e-4, top-10	λ=0.1	0.500	2.16
	λ=0.2	***0.503***	***2.67***
	λ=0.5	0.500	2.13
	λ=1	0.500	2.08
Avg. Growth Top-10			**2.26**
Max. Growth Top-10			**2.67**
a=256, lr=1e-4, top-100	λ=0.1	0.497	1.57
	λ=0.2	0.500	2.04
	λ=0.5	0.499	1.84
	λ=1	0.496	1.33
Avg. Growth Top-100			**1.70**
Max. Growth Top-100			**2.04**

Table 4. SFT a=512, lr=1e-4

Base Config	Variant	Aggregate Score	Growth, %
SFT a=512, lr=1e-4	—	0.494	—
a=512, lr=1e-4, top-10	λ=0.1	0.496	0.38
	λ=0.2	0.502	1.63
	λ=0.5	0.496	0.41
	λ=1	0.499	1.00
Avg. Growth Top-10			**0.86**
Max. Growth Top-10			**1.63**
a=512, lr=1e-4, top-100	λ=0.1	0.502	1.66
	λ=0.2	***0.503***	***1.83***
	λ=0.5	0.501	1.46
	λ=1	0.501	1.40
Avg. Growth Top-100			**1.59**
Max. Growth Top-100			**1.83**

Table 5. SFT a=512, lr=5e-5

Base Config	Variant	Aggregate Score	Growth, %
SFT a=512, lr=5e-5	—	0.488	—
a=512, lr=5e-5, top-10	λ=0.1	0.491	0.61
	λ=0.2	0.494	1.06
	λ=0.5	0.491	0.56
	λ=1	0.489	0.18
Avg. Growth Top-10			**0.60**
Max. Growth Top-10			**1.06**
a=512, lr=5e-5, top-100	λ=0.1	0.496	1.56
	λ=0.2	0.497	1.80
	λ=0.5	0.494	1.16
	λ=1	0.488	−0.14
Avg. Growth Top-100			**1.10**
Max. Growth Top-100			**1.80**

7. Limitations

While the proposed approach demonstrates consistent improvements in performance and efficiency over standard knowledge distillation baselines, several limitations remain that should be addressed in future work.

First, the observed performance gains are relatively modest (approximately 2–4% across benchmarks). Although the experiments show improvements, they suggest that further optimization of the distillation procedure is necessary to fully exploit the potential of cross-model knowledge transfer.

Second, the current study investigates only a single teacher–student configuration (`Qwen3-32B` → `Qwen3-4B`), as the 32B teacher model is currently the largest available model sharing the same tokenizer. Nevertheless, we believe that the proposed methodology is generalizable and can be applied to other model combinations, potentially leading to greater improvements.

Third, although the method is designed to improve training efficiency, we did not include quantitative measurements such as training throughput, total duration, or GPU-hour cost per epoch. Our analysis focused primarily on algorithmic efficiency and qualitative reductions in computational overhead (e.g., precomputed logits, use of LoRA). A more detailed profiling of hardware utilization and memory footprint will be presented in future work.

Finally, all experiments were conducted within a single computational environment and evaluated on a fixed set of Russian-language benchmarks described in Section 5.

Conclusion

This study evaluated the effectiveness of knowledge distillation (KD) relative to classical supervised fine-tuning (SFT) during the calibration phase of large language model adaptation for Russian. Our experiments, which distilled knowledge from a Russian-adapted RuadaptQwen3-32B teacher model into a 4B Qwen3 student, conclusively demonstrate that KD is a superior approach for enhancing model performance.

A key finding from our research is the fundamental trade-off between performance and computational efficiency, governed by the top-K hyperparameter. Incorporating a broader knowledge context from the teacher (top-100) consistently yielded superior results, achieving average performance gains of up to 3.27%. This performance, however, comes at a significant cost, requiring 62 GB of memory for the precomputed logits. Conversely, a top-10 configuration emerges as a viable, resource-efficient alternative, providing modest gains with a substantially smaller memory footprint of just 7 GB.

Our analysis also provides practical tuning guidelines. We found that a distillation weight of $\lambda = 0.2$ offers a robust and effective starting point across most configurations. Furthermore, we observed that the benefits of distillation are most pronounced for models with a lower LoRA α, indicating that simpler student models with less adaptive capacity gain the most from the teacher's guidance.

Our findings provide a practical roadmap for adapting large language models to morphologically complex languages like Russian, demonstrating that knowledge distillation effectively enhances the adaptation pipeline while emphasizing the importance of strategically balancing hyperparameters such as top-K, λ, and α with the available computational budget to achieve optimal performance.

Acknowledgements

The research was supported by the Russian Science Foundation, project No. 25-11-00191, `https://rscf.ru/project/25-11-00191/`.

The research was carried out using the MSU-270 supercomputer of Lomonosov Moscow State University.

References

1. Achiam, J., Adler, S., Agarwal, S., *et al.*: GPT-4 Technical Report. arXiv e-prints pp. arXiv–2303 (2023). `https://doi.org/10.48550/arXiv.2303.08774`

2. Anshumann, A., Zaidi, M.A., Kedia, A., *et al.*: Sparse logit sampling: Accelerating knowledge distillation in LLMs. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025. pp. 18085–18108. Association for Computational Linguistics (2025). `https://doi.org/10.18653/v1/2025.`

acl-long.885

3. DeepSeek-AI: Deepseek-v3 technical report (2024), https://arxiv.org/abs/2412.19437
4. Dubey, A., Jauhri, A., Pandey, A., *et al.*: The Llama 3 Herd of Models. arXiv e-prints pp. arXiv–2407 (2024). https://doi.org/10.48550/arXiv.2407.21783
5. Fenogenova, A., Chervyakov, A., Martynov, N., *et al.*: MERA: A Comprehensive LLM Evaluation in Russian. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 9920–9948. Association for Computational Linguistics (2024). https://doi.org/10.18653/v1/2024.acl-long.534
6. Goyal, N., Gao, C., Chaudhary, V., *et al.*: The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. Transactions of the Association for Computational Linguistics 10, 522–538 (2022). https://doi.org/10.1162/tacl_a_00474
7. Grashchenkov, P.V., Pasko, L.I., Studenikina, K.A., *et al.*: Russian parametric corpus RuParam. Journal Scientific and Technical of Information Technologies, Mechanics and Optics 158(6), 991 (2024). https://doi.org/10.17586/2226-1494-2024-24-6-991-998
8. Gu, Y., Dong, L., Wei, F., *et al.*: MiniLLM: Knowledge distillation of large language models. In: The Twelfth International Conference on Learning Representations, ICLR 2024. OpenReview.net (2024), https://openreview.net/forum?id=5h0qf7IBZZ
9. Gusev, I.: rulm: A toolkit for training neural language models (2023)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. Stat 1050, 9 (2015)
11. Hsieh, C.Y., Li, C.L., Yeh, C.K., *et al.*: Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes. In: Findings of the Association for Computational Linguistics, ACL 2023. pp. 8003–8017. Association for Computational Linguistics (2023), https://doi.org/10.18653/v1/2023.findings-acl.507
12. Hu, E.J., Shen, Y., Wallis, P., *et al.*: LoRA: Low-Rank Adaptation of Large Language Models. In: The Tenth International Conference on Learning Representations, ICLR 2022. OpenReview.net (2022), https://openreview.net/forum?id=nZeVKeeFYf9
13. Loukachevitch, N., Artemova, E., Batura, T., *et al.*: NEREL: A Russian Dataset with Nested Named Entities, Relations and Events. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2021. pp. 876–885. INCOMA Ltd. (2021), https://aclanthology.org/2021.ranlp-1.100
14. Loukachevitch, N., Tkachenko, N., Lapanitsyna, A., *et al.*: RuOpinionNE-2024: Extraction of Opinion Tuples from Russian News Texts. In: Proceedings of the International Conference "Dialogue". vol. 2025 (2025)
15. Men, X., Xu, M., Zhang, Q., *et al.*: ShortGPT: Layers in Large Language Models are More Redundant Than You Expect. In: Findings of the Association for Computational Linguistics, ACL 2025. pp. 20192–20204. Association for Computational Linguistics (2025), https://aclanthology.org/2025.findings-acl.1035/

16. Nikolich, A., Korolev, K., Bratchikov, S., *et al.*: Vikhr: Constructing a state-of-the-art bilingual open-source instruction-following large language model for Russian. In: Proceedings of the Fourth Workshop on Multilingual Representation Learning, MRL 2024. pp. 189–199 (2024). https://doi.org/10.18653/v1/2024.mrl-1.15

17. Ouyang, L., Wu, J., Jiang, X., *et al.*: Training language models to follow instructions with human feedback. In: Proceedings of the 36th Int. Conf. on Neural Information Processing Systems. vol. 35, pp. 27730–27744 (2022). https://doi.org/10.5555/3600270.3602281

18. Penedo, G., Kydlíček, H., Sabolčec, V., *et al.*: FineWeb2: One Pipeline to Scale Them All–Adapting Pre-Training Data Processing to Every Language. arXiv e-prints pp. arXiv–2506 (2025). https://doi.org/10.48550/arXiv.2506.20920

19. Raman, M., Mani, P., Liang, D., *et al.*: For distillation, tokens are not all you need. In: NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following (2023)

20. Team, Q.: Qwen3 technical report (2025), https://arxiv.org/abs/2505.09388

21. Tikhomirov, M., Chernyshev, D.: Facilitating large language model Russian adaptation with learned embedding propagation. Journal of Language and Education 10(4), 130–145 (2024). https://doi.org/10.17323/jle.2024.22224

22. Zhou, J., Lu, T., Mishra, S., *et al.*: Instruction-following evaluation for large language models. CoRR (2023). https://doi.org/10.48550/arXiv.2311.07911

Appendix

Table 6. Comparison of different hyperparameter configurations (a – LoRA alpha, lr – learning rate) across multiple evaluation benchmarks

a	lr	top	λ	mean	NEREL	Summ	MultiQ	USE	flores	copy	ru babilong	en IFEval	ru IFEval	en MMLU	ru MMLU	ru opinionne	ru param
128	1e-4	10	0.1	0.498	0.485	0.224	0.219	0.089	0.508	0.950	0.508	0.721	0.628	0.708	0.624	0.107	0.704
			0.2	0.497	0.494	0.226	0.223	0.113	0.509	0.940	0.512	0.719	0.627	0.709	0.621	0.104	0.665
			0.5	0.495	0.489	0.225	0.226	0.134	0.506	0.940	0.503	0.717	0.638	0.707	0.620	0.097	0.635
			1	0.495	0.479	0.224	0.226	0.162	0.505	0.945	0.498	0.708	0.612	0.708	0.622	0.105	0.644
		100	0.1	0.498	0.482	0.226	0.222	0.116	0.510	0.950	0.517	0.713	0.623	0.708	0.623	0.102	0.685
			0.2	0.498	0.488	0.225	0.225	0.126	0.508	0.945	0.515	0.713	0.634	0.706	0.623	0.100	0.666
			0.5	0.503	0.495	0.224	0.234	0.165	0.506	0.955	0.518	0.726	0.603	0.708	0.621	0.107	0.679
			1	0.495	0.494	0.225	0.230	0.161	0.503	0.960	0.500	0.702	0.603	0.708	0.619	0.093	0.636
256	1e-4	10	0.1	0.502	0.487	0.224	0.225	0.087	0.508	0.945	0.512	0.750	0.632	0.706	0.619	0.107	0.725
			0.2	0.503	0.496	0.225	0.221	0.117	0.509	0.955	0.515	0.708	0.645	0.707	0.621	0.108	0.709
			0.5	0.500	0.489	0.224	0.225	0.147	0.507	0.950	0.500	0.728	0.632	0.706	0.622	0.100	0.671
			1	0.500	0.492	0.226	0.219	0.170	0.505	0.960	0.493	0.726	0.619	0.706	0.623	0.101	0.657
		100	0.1	0.497	0.483	0.225	0.221	0.102	0.509	0.960	0.515	0.719	0.643	0.705	0.621	0.093	0.670
			0.2	0.500	0.492	0.224	0.221	0.155	0.508	0.960	0.507	0.721	0.623	0.706	0.622	0.107	0.649
			0.5	0.499	0.498	0.225	0.218	0.164	0.506	0.960	0.490	0.710	0.630	0.708	0.621	0.111	0.642
			1	0.496	0.490	0.225	0.230	0.159	0.504	0.965	0.498	0.702	0.621	0.705	0.616	0.099	0.636
512	1e-4	10	0.1	0.496	0.491	0.221	0.218	0.112	0.507	0.965	0.506	0.721	0.628	0.706	0.622	0.103	0.642
			0.2	0.502	0.510	0.227	0.222	0.109	0.508	0.960	0.522	0.730	0.619	0.707	0.622	0.093	0.693
			0.5	0.496	0.498	0.225	0.219	0.121	0.506	0.960	0.511	0.726	0.614	0.707	0.625	0.113	0.619
			1	0.499	0.495	0.225	0.225	0.151	0.507	0.960	0.506	0.726	0.617	0.706	0.622	0.112	0.630
		100	0.1	0.502	0.500	0.223	0.225	0.112	0.508	0.960	0.521	0.710	0.638	0.705	0.621	0.101	0.700
			0.2	0.503	0.502	0.224	0.223	0.125	0.509	0.960	0.507	0.728	0.638	0.705	0.620	0.118	0.676
			0.5	0.501	0.498	0.226	0.225	0.151	0.507	0.965	0.491	0.721	0.628	0.706	0.621	0.113	0.659
			1	0.501	0.490	0.228	0.234	0.167	0.503	0.960	0.507	0.726	0.610	0.707	0.619	0.106	0.650
	5e-5	10	0.1	0.491	0.482	0.225	0.216	0.056	0.506	0.955	0.508	0.719	0.623	0.707	0.622	0.115	0.653
			0.2	0.492	0.480	0.226	0.221	0.076	0.507	0.955	0.501	0.701	0.614	0.707	0.623	0.120	0.660
			0.5	0.491	0.468	0.228	0.220	0.104	0.507	0.960	0.506	0.715	0.628	0.706	0.620	0.104	0.618
			1	0.489	0.467	0.230	0.225	0.105	0.504	0.955	0.497	0.723	0.623	0.705	0.618	0.102	0.605
		100	0.1	0.496	0.487	0.224	0.220	0.073	0.507	0.960	0.513	0.713	0.628	0.705	0.619	0.113	0.685
			0.2	0.497	0.486	0.226	0.225	0.093	0.507	0.955	0.502	0.721	0.634	0.706	0.618	0.118	0.671
			0.5	0.494	0.474	0.229	0.222	0.102	0.506	0.960	0.503	0.717	0.617	0.705	0.617	0.120	0.646
			1	0.488	0.484	0.229	0.231	0.112	0.503	0.965	0.486	0.717	0.595	0.706	0.617	0.093	0.601