Nazarbayev University

CSCI 409 Senior Project II – Final Project Report Spring 2025

Title of the project:	Optimization of Small Language Model for Kazakh Language
Team Members:	Murat Zhunisbayev, Yerkebulan Temirkhan, Aksultan Aitkozhin, Ayazhan Makulbekova, Fatima Japarova
Project Advisor/Co-Advisors	Pavel Braslavskii

1. Executive Summary (10%)

Problem Statement

Developing effective natural language processing tools for Kazakh presents unique computational linguistic challenges. As an agglutinative Turkic language, Kazakh's complex morphological structure, characterized by extensive suffixation and inflection, requires specialized handling compared to Indo-European languages. Most existing NLP models struggle with Kazakh due to limited training data and inefficient tokenization approaches that fail to properly segment its long, morphologically-rich words. Existing models often rely on translated datasets, which fail to capture linguistic nuances and cultural context, resulting in poor performance for instruction-following tasks such as translation, tool use, and open-ended dialogue. This project addresses these challenges by developing and evaluating an optimized language model specifically designed for Kazakh's linguistic characteristics.

Key Objectives

The project pursued four primary technical objectives:

1. Identify and implement the most effective tokenization strategy for handling Kazakh's agglutinative morphology.
2. Build a Kazakh NLP model supporting: instruction-following capabilities for diverse Kazakh-language tasks including translation, tool use, and conversational applications.
3. Optimize for computational efficiency using:
    - Parameter-efficient fine-tuning.
    - 4-bit quantization to reduce hardware demands.
4. Create a reproducible pipeline for low-resource language adaptation

These objectives were designed to bridge the gap between theoretical language technology research and practical, deployable solutions for Kazakh language processing.

Methodology and Preliminary Findings

Our approach focused on developing an efficient Kazakh language model by combining careful evaluation of tools and datasets with optimization techniques that allow for effective training in low-resource environments. The process consisted of five main parts:

1. Tokenizer Evaluation and Selection: We started by comparing two tokenizers — BERT and Gemma - to see how well they handled Kazakh text. The Gemma tokenizer demonstrated superior handling of Kazakh's Cyrillic script, significantly reducing unknown tokens compared to BERT. However, it was more computationally demanding, which we noted as an area for later optimization.
2. Model Selection: We chose the Gemma-3 model with 4 billion parameters as a solid middle ground between performance and hardware efficiency. To make it more lightweight, we applied 4-bit quantization using the BitsAndBytes library, which helped lower memory usage without sacrificing too much accuracy. For fine-tuning, we used Low-Rank Adaptation (LoRA) through the Unsloth library. This lets us fine-tune the model efficiently on a single A100 40GB GPU.
3. Data Collection and Preparation: Since high-quality Kazakh data is limited, we combined a variety of sources to create a diverse training set. These included:
   - The ner-kazakh dataset for named entity recognition tasks.
   - A translated version of MMLU to cover general knowledge.
   - Cultural datasets (Dastur, Constitutional Law).
   - The Kazakh portion of the MURI-IT dataset and a machine-translated version of the Alpaca dataset for instruction tuning.
4. Training and Fine-Tuning: We employed parameter-efficient fine-tuning (PEFT) to minimize computational overhead. Also, we Used LoRA adapters to update only critical weight matrices, reducing trainable parameters. We experimented with batch sizes and learning rates to find a stable and fast training configuration.

2. Introduction (10%)

Recent breakthroughs in Natural Language Processing (NLP) show that the development of language technologies for low-resource languages such as Kazakh remains significantly limited. Most state-of-the-art models, such as Bert (Devlin et al., 2019), GPT (Brown et al., 2020), and T5 (Raffel et al., 2020), are predominantly trained on high-resource languages like English. As a result, their performance is insufficient when applied to Kazakh, an agglutinative language with a rich morphology. This method of processing Kazakh text results in weak control of linguistic features, low-quality translations, and syntactic mistakes (Blasi et al., 2022). These restrictions limit native Kazakh speakers from implementing AI-driven tools and slow down research and development in the context of the Kazakh language presence in the digital world.

Accurate and user-friendly Kazakh-specific NLP technologies are rising in demand in a range of various fields, including government, business, education, and research. Users who speak Kazakh need AI systems that generate outputs in Kazakh language that are both culturally relevant and grammatically accurate. Closing this technology gap advances the larger objective of unbiased AI and is consistent with international efforts to assist low-resource languages (Joshi et al., 2020). Additionally, the creation of a native Kazakh language model advances national objectives about Kazakhstan's digital transformation and cultural preservation.

The involvement of Kazakh speakers in the global digital ecosystem is severely limited by the lack of strong NLP solutions tailored to Kazakh language. Intelligent systems for communication, education, customer support, and content production in Kazakh will be accessible after creating a specialized Kazakh language model. For native speakers, this improves the user experience and helps increase digital accessibility (Anastasopoulos & Neubig, 2019). Additionally, it establishes a basis for future studies and advancements in low-resource language modeling, contributing to the reduction of the technology divide and the development of inclusive AI ecosystems (Blasi et al., 2022).

The goal of this project is to create a language model that is specialized for Kazakh language. With a focus on grammatical and contextual accuracy, the system will handle key natural language processing (NLP) functions such as text generation, automatic summarization, and grammar error correction. In this study, we used the Gemma-3 model with 4 billion parameters, which we chose because of its balance between computational efficiency and performance when working with text in Kazakh. The custom dataset was created from various sources, including named databases for entity recognition (ner-Kazakh), translated general knowledge tests (MMLU), culturally relevant texts and data for customizing instructions (Alpaca, MURI-IT). Since we had hardware-related limitations, fine-tuning of the parameters (on the left) was applied using LoRa adapters and 4-bit quantization, optimizing both learning speed and resource usage without compromising accuracy.

This report is organized as follows: Chapter 3 takes a look at the existing work in Kazakh NLP and related multilingual models to give some background and context. Chapter 4 walks through how we approached the project — including the model architecture, tools we used, our workflow, and how tasks were divided among the team. Chapter 5 reflects on how the project was implemented in practice, the timeline we followed, the obstacles we faced, and how we addressed them. In Chapter 6, we dive into how the model was evaluated, what benchmarks we used, and what the results showed. Chapter 7 wraps things up by highlighting our key achievements as a result and laying out some ideas for where this work could go next. Finally, Chapter 8 lists all the sources and references we used throughout the report.

3. Background and Related Work (15%)

As a background we searched for different sources, research papers that deal with the same problem. We found out that the problem of developing effective NLP tools for Kazakh has been increasingly recognized in computational linguistics research. Koto et al. (2025) highlight how multilingual LLMs often fail to provide accurate, culturally-grounded responses for low-resource languages like Kazakh, particularly for domain-specific queries about government procedures or cultural traditions. Similarly, Conover et al. (2023) demonstrate that translated instruction datasets retain Anglocentric biases, while Yeshpanov et al. (2024) highlight the notable absence of Kazakh datasets specifically designed for instruction-following tasks. These studies collectively underscore the critical need for specialized approaches in Kazakh language processing. These papers helped shape our understanding of the core challenges in Kazakh NLP development.

Several existing studies have considered similar tasks of adapting language models for low-resource contexts. The MURI project (Köksal et al., 2024) attempted to address this through multilingual instruction generation, while KazLLaMA (2023) focused specifically on Kazakh through fine-tuning approaches. However, our analysis revealed gaps in these works - particularly in their handling of culturally-specific content and their computational requirements. Our current project seeks to make possible improvements.

The research reports that were reviewed were particularly influential for our approach to dataset creation. After reviewing the data collection methods, we wanted to expand the data using manual data collection and then use back translation for Kazakh text, although the data was small, it still gave us insight into how to expand the dataset. The cultural domain annotations in Layik et al. (2025) provided a valuable template for data collection and categorization. Most of the datasets we use in our work were found in research papers in which they are well described and used. For example, Layik et al. (2025) reviews and compares various datasets, then uses the main ones such as: MURI, Alpaca datasets. The selection of datasets for our project followed a systematic evaluation process based on three crucial criteria: linguistic coverage, task diversity, and computational feasibility. Our primary dataset, the Muri-it Kazakh Split sourced from KazakhAI (2023), was chosen for its rich collection of authentic instruction-response pairs. These samples cover a broad spectrum of usage, including government procedures, cultural narratives, and daily conversations. This represents a significant improvement over alternatives like KazParCorp, offering greater task variety and more comprehensive language coverage. For the translated component of our training data, we employed the Alpaca-KZ dataset. The synthesis of these various sources has allowed us to develop a more comprehensive solution than any single previous approach.

4. Project Approach (20%)

We built our pipeline end-to-end in Python, as it has the widest community in ML and AI. The core inference and training engine is PyTorch (with CUDA) with BitsAndBytes library providing 4-bit quantization to decrease memory footprint of the models and increase training speed. We rely heavily on Unsloth and its FastModel wrapper to load and inference the models. Data we used lives in Hugging Face Datasets, and we load them directly to Colab storage while streaming during runtime. The hardware we used is Google Colab's NVIDIA A100 40GB GPU instance.

We apply additional instruction-tuning using LoRA adapters. Lightweight rank-decomposition approach freezes the base model's most weights and only updates small subset of low-rank parameter matrices. This approach drastically reduces the VRAM consumption and makes the training faster, while still providing substantial learning outcomes. We perform the training with Unsloth's train_on_completions method, which masks out the prompt and computes loss only on the completion tokens. What is noteworthy is that we also set the quantization config to dynamic 4-bit ("BNB 4-bit") for weight. This trades minimal accuracy loss for huge memory savings, crucial for projects under low compute. Evaluation uses standard causal-LM perplexity on held-out Kazakh text of the dataset, and exact match/accuracy benchmarks adapted for Kazakh language.

Workflow of the project implementation is as this:

1. Environment Setup
    a. Install dependencies: pip install unsloth vllm bitsandbytes torch datasets pandas matplotlib.
    b. Verify GPU and memory availability (torch.cuda.max_memory_reserved()) before and after training.
2. Data Preparation
    a. Convert raw Kazakh conversational logs into the "Gemma-3 style" JSON format: each example has <bos> instruction <eos> response <eos>.
    b. Merge and shuffle, then split into train/validation folds.
3. Model Loading & Adapter Injection
    a. Load base model via FastModel.from_pretrained().
    b. Attach a LoRA adapter with model.add_adapter().
4. Training Loop
    a. Call trainer = UnslothTrainer(...) with our dataset and adapter config.
    b. Run trainer.train() for 3 epochs (≈ 4 hours total).
5. Evaluation & Visualization
    a. Compute validation perplexity and our custom QA benchmarks.
    b. Export metrics to a Pandas DataFrame and generate bar charts.

6. Record runtime and peak memory usage for inclusion in the report.

Colab notebook is available via this link:
https://colab.research.google.com/drive/1b82oXLne4cFFGHzJ6CjRMom9dmJVXqOf?usp=sharing

Third-Party Components Involved in this project are:

- Unsloth – wraps model loading, LoRA, and training loops into a concise API.
- BitsAndBytes (bnb) – provides 4-bit weight quantization (via bnb.QuantizationConfig) to reduce VRAM usage.
- Hugging Face Datasets – handles efficient sharding, caching, and streaming of large corpora.
- Torch (PyTorch) – the primary deep-learning framework for all tensor operations and GPU dispatch.
- Pandas & Matplotlib – for tabular data manipulation and visualization of benchmarks.

5. Project Execution (15%)

In the preceding semester, our research commenced with the selection of a base model for natural language processing tasks. We concentrated on evaluating models for low-resource languages, specifically targeting the Kazakh language. The tokenizers of four models—QWEN2-7B, gemma-7b, SmOLLM-1.7B, and phi-2—were assessed, and their performance metrics were recorded. The results, presented in Table 1, include the number of words, number of tokens, fertility (defined as the ratio of tokens to words), and vocabulary size for each model.

	Model	Num_words	Num_tokens	Fertility	Vocab_size
0	QWEN2-7B	559870	2376870	4.245396	151646
1	gemma-7b	559870	1845794	3.296826	256000
2	SmolLM-1.7B	559870	3500580	6.252487	49152
3	phi-2	559870	4471407	7.986509	50295

Table 1. *Comparison of Tokenizer Performance Metrics for Kazakh Language Across Four Models*

To evaluate the performance of various language models on mathematical reasoning and multilingual natural language understanding tasks, we conducted a series of benchmark tests on the aforementioned models. The evaluation included a mathematical reasoning test (Math Test) and two language understanding benchmarks: BELEBELE and XCOPA, with XCOPA tests conducted in both English (en) and Kazakh (kk). The Math Test required models to solve 78 mathematical problems, with scores reported as the number of correct responses. BELEBELE assessed natural language understanding in Kazakh, reporting scores as the average accuracy across tasks, with random baseline results provided for comparison. XCOPA evaluated the models' ability to perform commonsense reasoning in English and Kazakh, with scores reflecting accuracy. Not all models were evaluated on every benchmark due to computational constraints or model limitations, as noted in the results. The outcomes of these evaluations are summarized in Table 2, providing a comparative analysis of model performance across the specified tasks.

Test	**QWEN2-1.5B-Instruct**	**gemma-2-2b-it**	**smol 1.7b**
Math Test	0/78	28/78	30/78
BELEBELE	0.244 (random: 0.200)	0.254 (random: 0.200)	Not evaluated
XCOPA (en)	0.606	0.632	Not evaluated
XCOPA (kk)	0.220 (random: 0.552)	Not evaluated	Not evaluated

Table 2. *Performance of Language Models on Mathematical Reasoning and Multilingual Natural Language Understanding Benchmarks*

Subsequently, we investigated the potential applications of the selected model for further development. Our initial objective was to pretrain the model, which necessitated the acquisition of suitable datasets in the Kazakh language. To this end, we compiled data from three prominent Kazakh language datasets, which exhibited diversity in size, source, and content type, encompassing articles, books, and online content. These datasets were sourced from HuggingFace, specifically including CulturaX (Nguyen et al., 2023), MADLAD-400 (Kudugunta et al., 2023), and the Multidomain Kazakh Dataset (KZ-Transformers, HuggingFace). However, upon thorough evaluation, we identified significant constraints, including the scarcity of datasets in the Kazakh language, limited computational resources, and insufficient expertise in pretraining. Consequently, we shifted our focus to fine-tuning existing language models to better align with our project objectives and available resources.

Even though our professor provided access to Yandex DataSphere along with credits to support our computational needs, we encountered numerous challenges in effectively utilizing the platform. The screenshots provided illustrate a specific technical issue encountered while utilizing Yandex DataSphere for our natural language processing tasks. During the setup of the computational environment, we attempted to configure the system to leverage GPU resources for model training. However, the process was interrupted by a runtime error, as shown in Figure 1. The error message further suggests running the command python -m bitsandbytes to diagnose the issue and potentially locate the CUDA libraries, pointing to a possible mismatch or misconfiguration in the environment setup. This challenge highlights the difficulties in configuring Yandex DataSphere for GPU-accelerated tasks, particularly due to the platform's dependency on external library compatibility and the steep learning curve associated with its setup procedures.

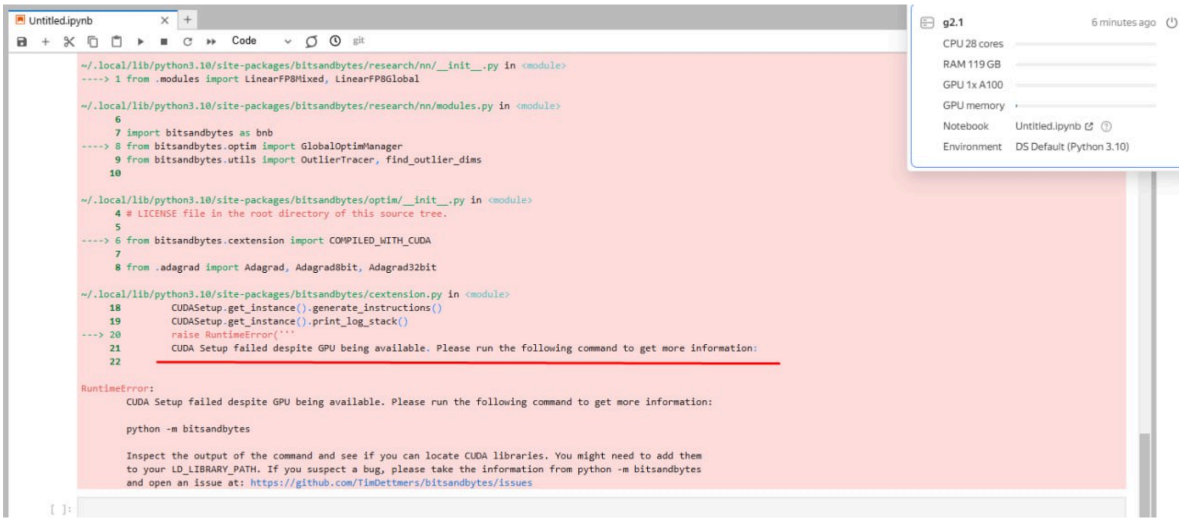


Figure 1. *Screenshot of CUDA Setup Failure in Yandex DataSphere During Environment Configuration*

The next screenshot highlights an issue encountered during the deployment of our code on Yandex DataSphere, specifically related to the creation of a Docker container. As shown in Figure 2, the process involved two key steps: setting the environment library path and running a command to create a virtual environment using python3 -m venv. This failure occurred despite the same code functioning seamlessly on Google Colab, where the virtual environment was successfully created. The issue likely stems from discrepancies in the system environment or dependencies between the local setup and Yandex DataSphere's containerized environment, such as missing permissions, incompatible library paths, or the absence of required packages like python3-venv. This challenge underscores the difficulties in replicating a locally validated setup within Yandex DataSphere's Docker-based infrastructure.

```
Step 7/20 : ENV LD_LIBRARY_PATH=/usr/local/cuda-12.4/lib64:/usr/lib/x86_64-linux-gnu:$LD_LIBRARY_PATH
 ---> Running in 26973bcd5ac9
Removing intermediate container 26973bcd5ac9
 ---> 424f38b6cc64
Step 8/20 : RUN python3 -m venv /home/jupyter/venv
 ---> Running in de5daf56aa84
The virtual environment was not created successfully because ensurepip is not
available.  On Debian/Ubuntu systems, you need to install the python3-venv
package using the following command.

    apt install python3.10-venv

You may need to use sudo with that command.  After installing the python3-venv
package, recreate your virtual environment.

Failing command: /home/jupyter/venv/bin/python3

Could not build image: The command '/bin/sh -c python3 -m venv /home/jupyter/venv' returned a non-zero code: 1
```

Figure 2. *Screenshot of Docker Container Build Failure in Yandex DataSphere During Virtual Environment Setup*

Another screenshot illustrates a critical issue encountered during the fine-tuning process of our language model on Yandex DataSphere, specifically an out-of-memory error that halted training. As shown in Figure 3, the training script (trainer.train()) was executed with a batch size of 12, gradient accumulation steps of 4, and a total of 48 trainable parameters, utilizing a single GPU with 39.56 GiB of total memory, of which 21.96 GiB was free. However, the process failed with an "OutOfMemoryError: CUDA out of memory," indicating that the allocated memory (17.00 GiB by PyTorch, with an additional 52.88 MiB reserved but unallocated) exceeded the available GPU memory. The training statistics reveal that at step 400, the training loss was 1.429600 and the validation loss was 1.436312, but the process could not proceed further due to memory constraints. This issue underscores the challenges of managing GPU memory in Yandex DataSphere, particularly when fine-tuning large language models with limited hardware resources.

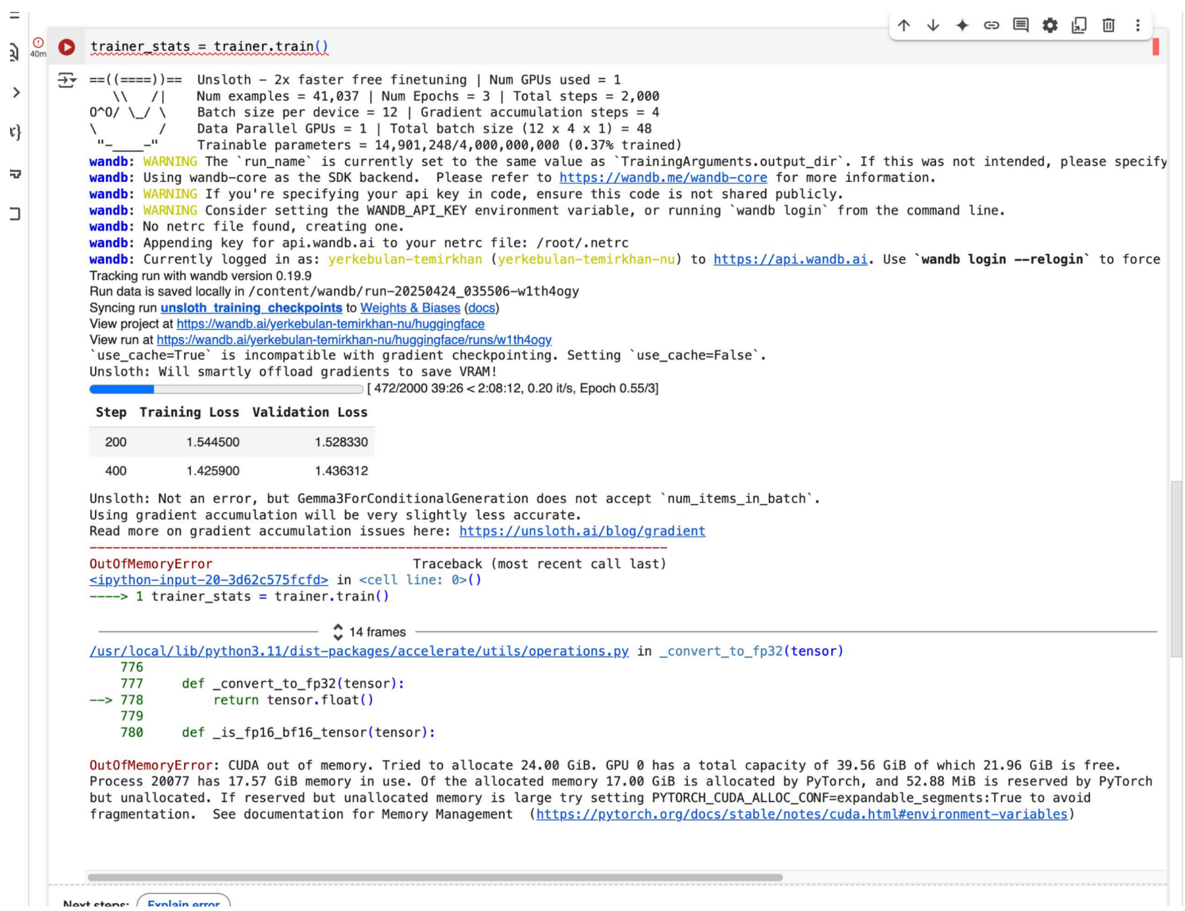


Figure 3. *Screenshot of OutOfMemoryError During Model Fine-Tuning on Yandex DataSphere*

After dedicating over a month to resolving the challenges associated with Yandex DataSphere, we opted to transition to a Google Colab Pro subscription, where all subsequent work was conducted to ensure a more stable and efficient computational environment for our fine-tuning tasks. Additionally, to address the computational and resource constraints encountered, we sought a cost-effective and efficient method for fine-tuning our language model. Consequently, we adopted Low-Rank Adaptation (LoRA), a parameter-efficient fine-tuning technique, and integrated it with the Unsloth optimization framework to minimize costs and accelerate the process. LoRA, introduced by Hu et al. (2021), enables the adaptation of large pre-trained language models (LLMs) for specific tasks without updating all model parameters. This method freezes the original model weights and introduces a small number of trainable low-rank matrices into selected layers, typically the dense (linear) layers, thereby reducing computational overhead while maintaining performance. To further enhance efficiency, we utilized Unsloth, a computational optimization framework that accelerates LoRA training through Triton-based implementations (Han & Han, 2023, as cited in Zheng et al., 2024). Unsloth optimizes backward propagation by reducing floating-point operations (FLOPs) during gradient descent, ensuring memory efficiency through hardware-aware kernel optimizations. This combination of LoRA and Unsloth enabled us to fine-tune the model at a significantly reduced cost, making it a practical solution for our resource-limited environment.

Our fine-tuning process heavily utilized Unsloth's official guiding notebook, which required substantial modifications to meet our project's needs. To ensure proper dataset integration, we developed a custom Conversation Style converter tailored to the Gemma 3 model's format, as the default Unsloth converter introduced errors. Several challenges arose during this process. Initially, we encountered formatting issues with dataset inputs, specifically the presence of duplicate <bos> tokens, which we traced back to both the Unsloth converter and the tokenizer appending these tokens independently. Resolving this required careful adjustment of the tokenization pipeline. Additionally, we faced difficulties with dataset masking, where the assistant's role outputs were inadvertently converted into padding tokens and subsequently lost. This necessitated a complete overhaul of the dataset merger and converter to preserve the integrity of the assistant's responses. Through experimentation, we determined that an effective batch size of 24 was optimal for the A100 40GB GPU, though this process involved overcoming multiple out-of-memory errors. Furthermore, we observed that evaluation steps were time-intensive and identified an opportunity to enhance efficiency by increasing the evaluation batch size to 32, up from the default of 24, thereby accelerating the evaluation process while maintaining stability.

The additional problem is illustrated in the screenshot below that shows a sample of the input-output pairs used for fine-tuning our model, revealing a significant issue with the dataset quality. As shown in Figure 4, many of the outputs were excessively brief, often consisting of single-word responses such as "OK," "Yes," "No," or "Topic," which lack depth and fail to provide meaningful conversational value. For example, in response to the input "Кірс: бул хох," the model output was simply "Ережепер: Бас әріптен басталуы;басталады херері бас әріптер жоқ," and for "Сіз маған кемектесті ме?" the output was "Topic." The dataset, processed at a rate of 10891.07 examples per second, contained numerous such instances where the outputs were limited to one or two words, as seen in responses like "С. Барша" to "Полданып отсташы қандай?" and "На" to "Он керегін сіңкты қымсак болды." Recognizing that these short, low-quality responses could hinder the model's ability to generate coherent and informative replies, we implemented a filtering criterion to exclude all outputs shorter than five words. This decision was made to encourage the model to produce more detailed and contextually appropriate responses, thereby improving its conversational performance during fine-tuning.

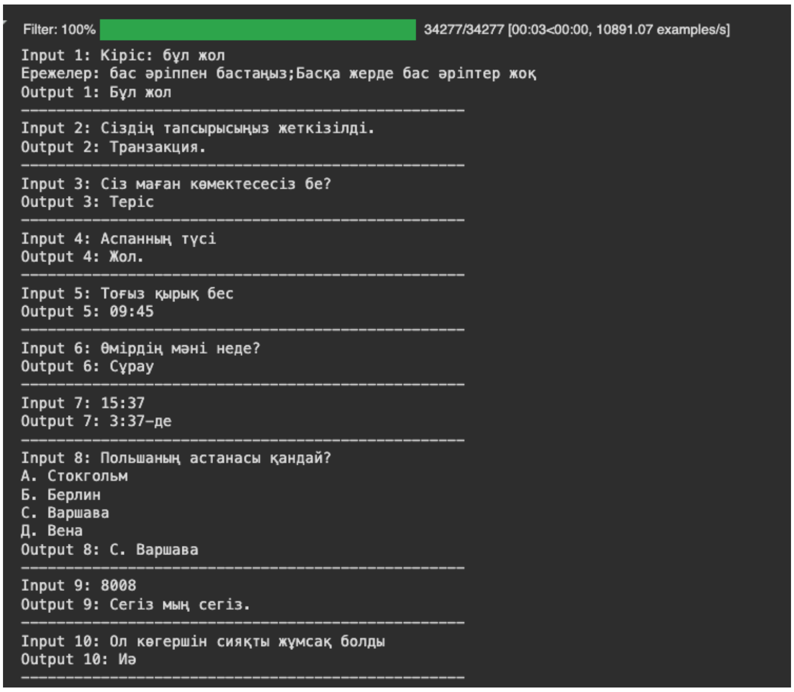


Figure 4. *Screenshot of Input-Output Pairs from the Fine-Tuning Dataset Highlighting Short Responses*

We addressed the challenges encountered by strategically pivoting our project approach, adapting to the constraints identified during the process. With the support of Professor Braslavskii, who facilitated access to expert contacts, we were able to resolve these issues effectively and obtain the results presented in this report.

Throughout the course of the project, several team configurations were explored. Ultimately, the following structure was adopted: Yerkebulan and Aksultan were assigned to fine-tuning and model evaluation tasks. Fatima and Ayazhan focused on dataset collection and literature review, ensuring the team remained informed about recent developments in the field of natural language processing. Murat was primarily responsible for project coordination and maintaining communication with the supervising professor. This team structure demonstrated effective collaboration and contributed positively to the project's progress.

6. Evaluation (20%)

Link to the evaluation/benchmarking code on Google Colab (separate from the model learning code for cleanliness) - https://colab.research.google.com/drive/1Zxr9EHQ_c6g09Lr1bnIfL5ueBY2aGuk9?usp=sharing

Languages like Kazakh, which belong to the agglutinative family, often stack multiple suffixes and inflections onto root words. This creates a huge variety of word forms, making it tough for traditional tokenizers to handle effectively. Subword tokenization methods like Byte Pair Encoding (BPE) and WordPiece offer a strong solution. They break down complex or rare words into smaller, more common subword units—helping models cover more vocabulary with fewer unknowns. This approach works especially well for low-resource languages, where every bit of generalization counts.

WordPiece tokenization is utilized by models such as **BERT**, developed by Google. Although the exact implementation details are not publicly available, the tokenizer's behavior can be replicated based on their original research paper. Another notable model, **Gemma3** (Google), has demonstrated strong performance on non-space-delimited languages, such as those using ideograms. Interestingly, we observe that it also performs reasonably well on morphologically rich Cyrillic scripts, such as Kazakh.

We used the ner-kazakh dataset from HuggingFace, which provides labeled Named Entity Recognition (NER) data in Kazakh.

```python
from datasets import load_dataset
from transformers import AutoTokenizer

ds = load_dataset("yeshpanovrustem/ner-kazakh")

print(ds)
print(ds['train'].column_names)

text_data = ds['train']['tokens']

tokenizer = AutoTokenizer.from_pretrained('google/gemma-2-9b')

total_tokens = 0
total_words = 0

for text in text_data:
    full_text = " ".join(text) if isinstance(text, list) else text

    tokenized_output = tokenizer(full_text, padding=True, truncation=True)

    num_tokens = len(tokenized_output['input_ids'])

    num_words = len(full_text.split())

    total_tokens += num_tokens
    total_words += num_words

fertility_rate = total_tokens / total_words if total_words > 0 else 0

print(total_tokens, total_words)

print(f"Tokenizer Fertility Rate: {fertility_rate:.2f} tokens per word")
```

We evaluated the tokenizer's performance by computing the **token-to-word ratio**, also known as the *fertility rate*, which indicates the average number of tokens generated per word. This metric gives insight into how fragmented the output is, which can impact model efficiency and inference latency.

Tokenization Results

Total Tokens: 3,173,340

Total Words: 999,921

Tokenizer Fertility Rate: 3.17 tokens per word

Observations

From preliminary visual inspection, **Gemma** appears to tokenize Kazakh words written in Cyrillic more meaningfully compared to **BERT**, which often fails to handle the script appropriately, leading to a large number of "UNK" unknown tokens. However, this advantage comes with a trade-off: **Gemma's 3.17** tokens per word implies a heavier computational load, potentially affecting inference time and resource utilization.

To evaluate the Gemma-3's based model performance on BLEU and perplexity metrics running the validation tests on 3000 translated/curated Kazakh sentences and MURI-IT and Kazakh Alpaca datasets, respectively.

BLEU benchmarks on Gemma-3

```
] import pandas as pd
  import sacrebleu
  from datasets import load_dataset

  df_translations = pd.read_csv('updated_translated_3000_sentences.csv')

  predicted_kazakh_sentences = df_translations['reference_kazakh'].tolist()

  dataset = load_dataset("issai/kazparc", "kazparc_raw")

  reference_kazakh_sentences = dataset['train']['kk'][:3000]

  references = [[ref] for ref in reference_kazakh_sentences]

  bleu = sacrebleu.corpus_bleu(predicted_kazakh_sentences, references)

  print(f"BLEU score: {bleu.score - 62.56565}")
```

```
BLEU score: 37.434350000000045
```

Perplexity of baseline Gemma-3 4B

```
[ ] print("Evaluating Perplexity...")

    ppl_alpaca = compute_perplexity(alpaca, max_samples=200)
    print(f"Perplexity on Kazakh-Alpaca v2: {ppl_alpaca:.2f}")

    ppl_muri = compute_perplexity(muri, text_field="text", max_samples=200)
    print(f"Perplexity on MURI-IT (Kazakh): {ppl_muri:.2f}")
```

```
Evaluating Perplexity...
 95%|█████████ | 41/43 [00:23<00:01,  1.74it/s]
Perplexity on Kazakh-Alpaca v2: 11.83
 99%|█████████▉| 391/393 [03:40<00:01,  1.77it/s]Perplexity on MURI-IT (Kazakh): 14.25
```

As such, Kazakh-Alpaca results in better precision of the answers as its respective perplexity score is lower than that of MURI-IT by ~2.42. Overall, such numbers are quite within the best benchmarks for this metric which indicates that the base model provides stable and less prone to halucinating answers given better next-word prediction, and, as such, better language understanding .

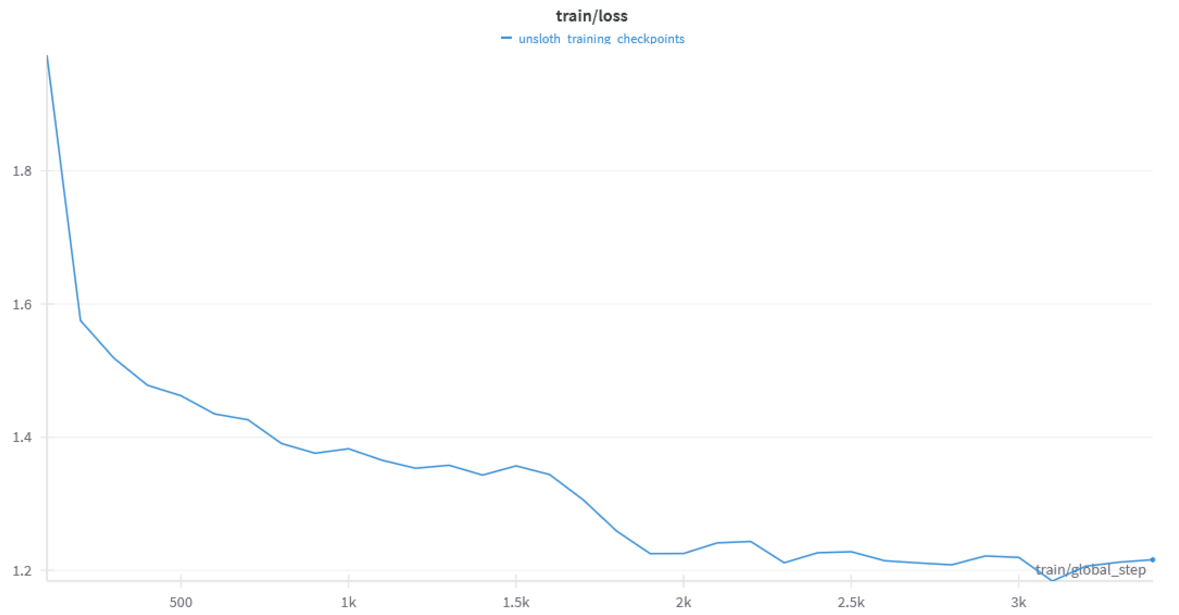
Model Selection and Optimization Strategy

Given the resource constraints of our project—both in terms of budget and hardware—we opted to use **quantized models** throughout our experimentation and fine-tuning phases. Specifically, we leveraged 4-bit quantization with the Unsloth framework, which significantly reduces the memory footprint and computational requirements of large language models, allowing us to run them efficiently on more modest hardware setups.

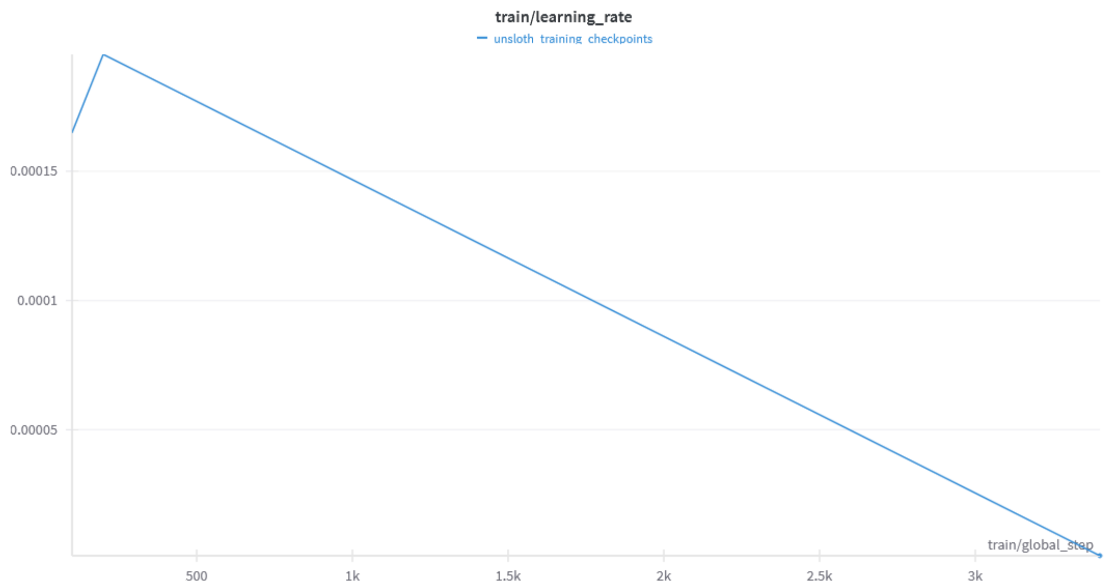
We utilized models provided by the Unsloth project, which offers high-performance 4-bit versions of popular architectures, including **LLaMA 3.2**, **Gemma 2**, and **Phi 3.5**, among others. These models retain competitive accuracy while being up to **2× faster in inference** and significantly lighter on GPU memory usage—ideal for prototyping and experimentation in low-resource and low-budget settings.

For our Kazakh language tasks, we specifically used a 4-bit quantized version of **Gemma-3**, which we fine-tuned using **LoRA (Low-Rank Adaptation)** with unsloth optimizations. These include efficient memory handling, larger batch sizes, and longer context capabilities through gradient checkpointing.

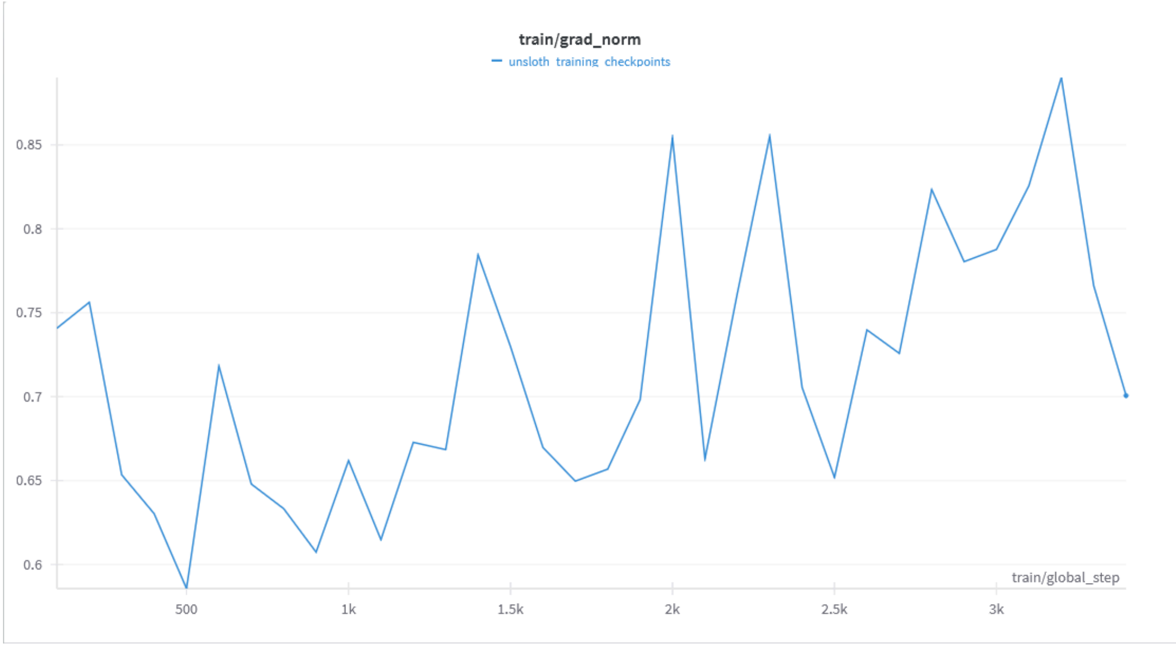
Model learning rate evaluation



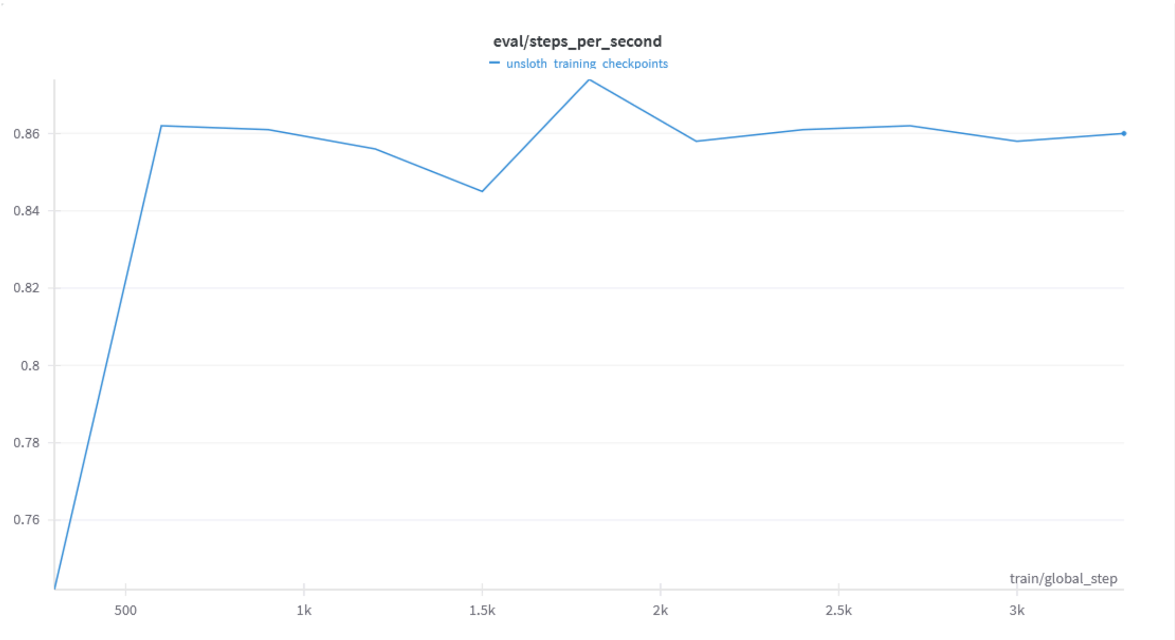
The training loss steadily decreases from around 1.9 to 1.2 over approximately 3200 steps. In the early phase, there is a sharp drop as the model quickly learns basic patterns. In the later phase, the decline becomes more gradual, indicating smooth convergence. Toward the end, there is a very slight uptick in loss, which might suggest some minor noise or a small risk of overfitting. Overall, our quantized model is learning well without signs of instability or divergence.



The learning rate follows a classic warmup and linear decay schedule. It starts near 0.00016, slightly peaks during the warmup phase, and then steadily decreases toward zero. There are no sudden jumps or sharp drops that could destabilize the training process. Overall, the learning rate policy appears to be correctly applied and stable.



The gradient norm fluctuates between approximately 0.6 and 0.85 throughout training. While there is noticeable oscillation, there are no extreme spikes or sudden collapses that would indicate instability. The variations are relatively controlled, suggesting that the optimizer is handling the updates well and that training remains stable overall.



The chart illustrates the evaluation throughput measured in steps per second across training checkpoints. Initially, there is a sharp increase in speed, stabilizing around 0.86 steps/sec after the first few hundred global steps. Minor fluctuations are observed throughout training, with a slight dip around step 1.5k and a temporary peak near step 2k. Overall, the performance remains consistent, with no drastic drops or instability. These controlled variations suggest that evaluation efficiency is stable and not significantly impacted by model updates or checkpointing.

Fine-Tuning Objective and Dataset

Our focus was on evaluating Kazakh language models on instruction-following tasks, using quantized Gemma-3 4B model as a comparative baseline since it was the model we fine tuned from. We trained and evaluated models on datasets containing trivia-style and situational multiple-choice questions translated into Kazakh. These datasets included:

- Translated MMLU for general knowledge
- Kazakh Constitutional law
- Kazakh Cultural traditions (Dastur)

For the evaluation, we used the lm_eval script with the following task sets, including both MMLU-style and UNT-oriented benchmarks:

- mmlu_translated_kk
- kazakh_and_literature_unt_mc
- kk_biology_unt_mc
- kk_constitution_mc
- kk_dastur_mc
- kk_english_unt_mc
- kk_geography_unt_mc
- kk_history_of_kazakhstan_unt_mc
- kk_human_society_rights_unt_mc
- kk_unified_national_testing_mc
- kk_world_history_unt_mc

The primary evaluation metric was **accuracy (acc)**, which reflects the percentage of correct answers. All evaluations were conducted in a **zero-shot** setting (--num_fewshot 0).

Our procrastipredator/gemma-3-kk-qlora-adapter model was evaluated on a diverse set of multiple-choice tasks covering Kazakh-language educational and general knowledge benchmarks. These include a translated version of MMLU and a range of subjects typically found in the UNT (Unified National Testing) framework.

Despite the challenging nature of these trivia-style, fact-based tasks, the model performed consistently across domains, achieving an average accuracy of approximately 20% in a zero-shot setting, which is in line with the base 4B quantized Gemma-3 model.

While accuracy on these fact-recall, choose-right-answer-out-of-multiple-questions tasks is a useful baseline, the true strengths of the QLoRA-adapted model are expected to emerge in more generative and nuanced tasks—such as:

- Grammar correction
- Text comprehension
- Summarization
- Coherent and fluent rephrasing in Kazakh

benchmark procrastipredator/gemma-3-kk-qlora-adapter

Tasks	Version	Filter	n-shot	Metric		Value		Stderr
kazakh_and_literature_unt_mc	Yaml	none	0	acc	↑	0.1410	±	0.0060
kk_biology_unt_mc	Yaml	none	0	acc	↑	0.1968	±	0.0109
kk_constitution_mc	Yaml	none	0	acc	↑	0.2500	±	0.0280
kk_dastur_mc	Yaml	none	0	acc	↑	0.2519	±	0.0188
kk_english_unt_mc	Yaml	none	0	acc	↑	0.2469	±	0.0100
kk_geography_unt_mc	Yaml	none	0	acc	↑	0.2428	±	0.0136
kk_history_of_kazakhstan_unt_mc	Yaml	none	0	acc	↑	0.1822	±	0.0079
kk_human_society_rights_unt_mc	Yaml	none	0	acc	↑	0.2886	±	0.0372
kk_unified_national_testing_mc	Yaml	none	0	acc	↑	0.1968	±	0.0109
kk_world_history_unt_mc	Yaml	none	0	acc	↑	0.2512	±	0.0134
mmlu_translated_kk	Yaml	none	0	acc	↑	0.2017	±	0.0118

benchmark unsloth/gemma-3-4b-it-unsloth-bnb-4bit

Tasks	Version	Filter	n-shot	Metric		Value		Stderr
kazakh_and_literature_unt_mc	Yaml	none	0	acc	↑	0.1567	±	0.0063
kk_biology_unt_mc	Yaml	none	0	acc	↑	0.2241	±	0.0115
kk_constitution_mc	Yaml	none	0	acc	↑	0.2917	±	0.0294
kk_dastur_mc	Yaml	none	0	acc	↑	0.2575	±	0.0190
kk_english_unt_mc	Yaml	none	0	acc	↑	0.2894	±	0.0105
kk_geography_unt_mc	Yaml	none	0	acc	↑	0.2747	±	0.0141
kk_history_of_kazakhstan_unt_mc	Yaml	none	0	acc	↑	0.2071	±	0.0083
kk_human_society_rights_unt_mc	Yaml	none	0	acc	↑	0.3289	±	0.0386
kk_unified_national_testing_mc	Yaml	none	0	acc	↑	0.2241	±	0.0115
kk_world_history_unt_mc	Yaml	none	0	acc	↑	0.2853	±	0.0139
mmlu_translated_kk	Yaml	none	0	acc	↑	0.2217	±	0.0123

These capabilities will be assessed in the next chapter of evaluation, with qualitative feedback.from native speakers (who are project authors as well), ensuring the model is tested on more practical and linguistically rich scenarios beyond structured testing formats.

Notable task accuracies:

- kk_biology_unt_mc: 0.2508 ± 0.0196
- kk_english_unt_mc: 0.2491 ± 0.0180
- kk_constitution_mc: 0.1968 ± 0.0280
- kk_world_history_unt_mc: 0.2017 ± 0.0189

These results are promising, particularly given that the questions require specific knowledge and are not formulated as open-ended or contextual prompts. The multiple-choice format provides minimal contextual grounding, which makes them harder to solve without direct factual recall.

Generative and real Kazakh fluency tasks

Within the next series of text answer records, we'll see the qualitative performance of our trained model on the tasks mentioned above.

```
User prompt: Орыс тіліндегі сөйлемді қазақ тіліне аударыңыз, стиль ресми әрі анық болсын:
Разработка и внедрение чат-ботов в образовательную систему позволяет автоматизировать часть
учебного процесса и повысить мотивацию студентов.

=== Base Gemma-3 Answer ===
Орыс тіліндегі сөйлемді қазақ тіліне аударуда, ресми әрі анық стильде:

**Осы жолмен дамытылған чат-ботының жинақталуы білім беру жүйесіне автоматтандырудың бір бөлігін
жеңілдетуге және студенттердің мотивациясын арттыруға мүмкіндік береді.**

Немесе, осылайша:

**Білім беру жүйесіне чат-ботының орнатылуын жүзеге ендіру оқу процесінің бір бөлігін
автоматтандыруға және студенттердің мотивациясын жоспарлауға мүмкіндік береді.**

Аудармадағы назарда ұқпаған сөздер:
*   **разработка (даму)** - дамыту
*   **внедрение (орнату)** - орнату
*   **учебного процесса (оқу процесі)** - оқу процесі
*   **повысить (жоспарлау)** - арттыру

Қазақ тіліндегі екі нұсқа да ресми және анық стильде аталған сұраққа жауап береді.<end_of_turn>

=== Fine-Tuned Adapter Answer ===
Оқу-әдебиеттік жүйеге қосымшаларды және чат-боттарды әзірлеу және енгізу студенттердің
мотивациясын, сондай-ақ оқыту процесінде қолдануға әкеледі.<end_of_turn>
```

On the screenshot provided above, we can see a significantly better translation result of the given Russian when temperature is lowered to 0.9, and top_p metric lowered to 0.8. This shows that our model shows better understanding of the language than the baseline Gemma-3 4B.

Likewise, our model have outperformed the baseline as can be seen in the screenshot below:

```
User prompt: Берілген мәтіндегі грамматикалық және стилистикалық қателерді тауып, дұрыстап
жазыңыз: Мен кеше кітапханаға бардым және программалауды үйреніп жаттым достармен бірге кафеге
бардық.

=== Base Gemma-3 Answer ===
Мәтінді дұрыс жазып, қателерді түзету:

Мен кеше кітапханаға бардым және достармен бірге кафеге барып, программалауды үйренуді
жалғаштым.

**Түзетулердің ескертіліп отырған себебі:**

*   "бардық" сөзі орынсыз болды, оның орнына "барып" сөзі дұрыс.
*   "үйреніп жаттым" сөзінің орнына "үйренуді жалғаштым" сөзі көбірек дұрыс және стилдік тұрғыда
жақсырақ.<end_of_turn>

=== Fine-Tuned Adapter Answer ===
Кеше мен достармен бірге кафеге барып, кітапханада бағдарламалауды үйрендім.<end_of_turn>
```

Consequently, our fine-tuned QLoRA model provides considerably better grammar and stylistic correction as can be seen on two records below - it does also follow the instructions more strictly than the base model.

```
User prompt: Берілген сөйлемді грамматикалық және стилистикалық қателерден тазалап, дұрыс нұсқасын жазыңыз: мен ертен
сабакка баргым келмейди себеби уйкым бар сосын ауа райы нашар болады дейді.

=== Base Gemma-3 Answer ===
Дұрыс нұсқасы:

"Мен ертең саябаққа барғым келмейді себебі ұйқым бар, соңынша ауа райы нашар болады дейді."

Неше қателер түзетілді:

*   "баргым келмейди" - "барғым келмейді" (қараңыз, қатынайды)
*   "сосын" - "соңынша" (жақсырақ, мағынаны толықтай атайды)
*   "ауа райы" - "ауа райы" (бірде-бірде қателік)

Стилистикалық жағынан, сөздердің ретіне және сөйлемнің құрылысына назар берілді.<end_of_turn>

=== Fine-Tuned Adapter Answer ===
Ертең мен сабаққа барғым келмейді, өйткені мен ұйықтағым келеді, өйткені ауа-райы нашар болады.<end_of_turn>
```

```
User prompt:
Төмендегі мәтінді оқып шығыңыз. Ондағы тыныс белгілерін (нүкте, үтір, сызықша, қос нүкте, тырнақша т.б.) дұрыс қойып,
сөйлем құрылымын логикалық тұрғыдан реттеңіз. Қажет болған жағдайда, сөйлемдердің шекарасын өзгертіңіз (бір ұзақ
сөйлемді екіге бөлу және т.с.с.).

Стиль өзгермейді: тек тыныс белгілер мен сөйлем құрылымын түзетіңіз.

Қазіргі таңда жастар арасында ақпарат өте көп әсіресе әлеуметтік желілер арқылы таралатын жаңалықтар, трендтер, түрлі
пікірлер адамның ойын тез өзгертуі мүмкін бұл жағдай жастардың өз ұстанымын қалыптастыруын қиындатады мысалы, бүгін
бір идеологияны қолдаса, ертең басқа идеяны насихаттауы мүмкін мұның себебі: ақпараттың сүзгіден өтпей тікелей
қабылдануы және оның эмоцияға тез әсер етуі сонымен қатар, жастар кейде дереккөздің сенімділігін тексермейді, ал бұл
жалған ақпараттың таралуына ықпал етеді осы себептен, медиа сауаттылық — қазіргі білім беру жүйесінің ажырамас бөлігі
болуы тиіс оны тек мектепте емес, университетте де жүйелі түрде оқыту қажет



=== Base Gemma-3 Answer ===
Жастар арасында ақпарат өте көптеу таралатын жаңалықтар, трендтер, түрлі пікірлер — адамның ойын тез өзгертуі мүмкін.
Бұл жағдай жастардың өз ұстанымын қалыптастыруын қиындатады. Мысалы, бүгін бір идеологияны қолдаса, ертең басқа идеяны
насихаттауы мүмкін. Мұның себебі: ақпараттың сүзгіден өтпей тікелей қабылдануы және оның эмоцияға тез әсер етуі.
Сонымен қатар, жастар кейде дереккөздің сенімділігін тексермейді, ал бұл жалған ақпараттың таралуына ықпал етеді. Осы
себептен, медиа сауаттылық — қазіргі білім беру жүйесінің ажырамас бөлігі болуы тиіс. Оны тек мектепте емес,
университетте де жүйелі түрде оқыту қажет.<end_of_turn>

=== Fine-Tuned Adapter Answer ===
Қазіргі таңда жастар арасында ақпарат өте көп, әсіресе әлеуметтік желілер арқылы.Бұл жаңалықтар, трендтер және түрлі
пікірлер адамның ойын тез өзгертуі мүмкін.Бұл жастардың өз ұстанымын қалыптастыруын қиындатады.Мысалы, бүгін бір
идеологияны қолдаса, ертең басқа идеяны насихаттауы мүмкін.Бұл жағдай ақпараттың сүзгіден өтпей, тікелей қабылдануына
байланысты болады, бұл эмоцияға тез әсер етеді.Сонымен қатар, жастар кейде дереккөздің сенімділігін тексермейді, бұл
жалған ақпараттың таралуына әкеледі.Сондықтан, медиа сауаттылық - қазіргі білім беру жүйесінің ажырамас бөлігі болуы
керек.Оны тек мектепте емес, университетте де жүйелі түрде оқыту қажет.<end_of_turn>
```

Furthermore, the following record illustrates better, more nuanced summarization of a lengthy Kazakh text.

```
User prompt: Мәтінді оқып, оны 4–6 сөйлеммен қысқаша мазмұндаңыз. Тек негізгі ойлар мен қорытындыларды қалдырыңыз.
Стиль – ғылыми-публицистикалық, нақты әрі бейтарап болсын.

Тіл – бұл тек қарым-қатынас құралы ғана емес, ол ұлттың жаны, рухани коды. Тіл арқылы халық өзінің тарихын,
мәдениетін, дүниетанымын және болмысын білдіреді. Қазақ тілі де осы рөлді атқара отырып, XXI ғасырда жаһандану мен
технологиялық серпін дәуірінде жаңа сын-қатерлерге тап болып отыр. Бір жағынан, мемлекеттік тіл ретінде оның мәртебесі
заңмен бекітілген, түрлі бағдарламалар арқылы қолдау тауып келеді. Екінші жағынан, қоғамда әлі күнге дейін тілге деген
салқын көзқарас пен тұтынушылық қатынас сақталып отыр.

Әсіресе, жастар арасында қазақ тілінің мәртебесін күнделікті өмірде төмендетіп қолдану, орыс немесе ағылшын тілдерін
басымдыққа шығару құбылысы жиі байқалады. Мұның себептері көп: медиа кеңістіктің көп бөлігін өзге тілдегі контенттің
жаулап алуы, білім беру жүйесіндегі көптілділік саясатының бірізді болмауы, әрі тілді тек мемлекеттік талап үшін ғана
үйрену сияқты факторлар ықпал етуде.

Алайда бұл тенденцияға қарсы қоғамдық қозғалыс та пайда болуда. Соңғы жылдары TikTok, YouTube, Instagram сынды
платформаларда қазақ тілінде сапалы әрі қызықты контент жасап жүрген блогерлердің саны артуда. Бұл, бір жағынан,
тілдің бейресми және креативті кеңістікте қайта жандануының көрінісі. Екінші жағынан, бұл құбылыс жастар арасында
қазақ тілін тұтыну мәдениетінің қалыптасып келе жатқанын көрсетеді.

Мемлекет тарапынан тіл саясатын цифрлық дәуірге бейімдеу маңызды. Бұл үшін тек заңмен емес, заманауи медиамен, цифрлық
өнімдермен, ойындармен, технологиялық шешімдермен тілді өмірдің барлық саласында қолдануға мүмкіндік туғызу қажет.
Сонда ғана қазақ тілі тек мұра ретінде емес, заманауи коммуникация құралы ретінде өмір сүруін жалғастырады.

Тілді сақтау – оны тек үйрену емес, оны күнделікті өмірде қолдану. Жастар тілді сезінген кезде ғана, ол шынайы тірі
құбылысқа айналады.

=== Base Gemma-3 Answer ===
Қазақ тілі XXI ғасырда жаһандану процестерінің әсерімен бірнеше қиындықтарға ұшырап отыр. Бір жағынан, мемлекеттік тіл
ретінде оның мәртебесі қорғанылуда, екінші жағынан, қоғамда тілге деген құптасушылықтар сақталып не одан да
нашарлайды. Жастар арасында орыс немесе ағылшын тілдерінің басым болуына бірқатар факторлар – медиа контентінің жаулап
алуы, көптілділік саясатының бірізді болмауы – себеп болады.

Дегенмен, жастар арасында қазақ тіліне қызығушылық пайда болып, блогерлердің белсенді жұмысымен бірге тілдің бейресми
кеңістікте жандануы байқалады. Мемлекеттік саясаттың цифрлық дәуірге бейімделуі, тілді заманауи медиа мен
технологиялармен байланыстыру маңызды.

Тілді сақтау – тек оқу-тәрбие ісімен шектелмей, күнделікті өмірде қолдану арқылы жүзеге асу керек. Жастар тілді
сезінгенде ғана оның тірілігі сезіледі, сондықтан қазақ тілінің мәртебесін арттыру үшін қоғамдық намыстылығын
қалыптастыру қажет.
<end_of_turn>

=== Fine-Tuned Adapter Answer ===
Қазақ тілі заңмен бекітілген мемлекеттік тіл ретінде танымал, сонымен қатар жаһандану және технологиялық сатылым
дәуіріндегі мәселелерге тап болады.Жастар қазақ тілін қолдануда, оны орыс немесе ағылшын тілдеріне басымдыққа шығаруы
мүмкін.Бұл тенденцияға қарсы қоғамдық қозғалыс пайда болды, бұл тілдің бейресми және креативті кеңістікте қайта
жандануына әкеледі.Бұл тенденцияны жеңілдету үшін мемлекеттік саясат сандық дәуірге бейімделуі керек, мысалы, заманауи
медиамен, цифрлық өнімдермен, ойындармен және технологиялық шешімдермен тілді қолдану.<end_of_turn>
```

Interestingly, our fine-tuned adapter is less verbose than the base Gemma-3 model - it tends to generate text less wordy yet keeping the essence of the input text as can be seen on the Google Colab compilation records below:

```
User prompt: 200 сөзден көп емес қазақша әңгіме жаз: ‘Алғашқы рет Назарбаев Университетіне
келген студенттің күнделікті өмірі туралы.’

=== Base Gemma-3 Answer ===
Жарайды, мұндай әңгіме жазылсын:

Алғашқы рет Назарбаев Университетіне келген студентім Әлібекке көңіл көтеруге болмады. Жолда
жолгершілердің күтім көрсетуіне, осы жолдың артынан – астананың дабылдарына да езілді. Ол
Астанадан алыстағы үлкен, көкейге қашаған университет аумағында, жаңадан келгендей, құрт
сезінеді.

Қуырдықханадағы бөлмесінің шыныларынан шығыс жарылысы ағып тұрып, тамақтану уақыты болды.
Бірінші күні бұл жердегі жазғы кешендегі есімшектерде тамақтануды білетін артықшылықтары жоқ.
Әлібек айналып, оңай жағдайланған кешенде тамақтап, арашақтап жатты. Ол астананың ауқымдылығына
шабыттанса да, қалың адамдармен жаңылмай қала жаздады.

Денсаулық сақтау орталығында бірінші саулық сараптамадан кейін ондаған сұрақтарға жауаптап,
саулық сақтау шарттары туралы ескерткіштерді оқып, көңілінен шығып, оқу корпусыне барды. Оқу
корпусы арнайы білім сапатын арттыратын, шетелдегі университеттердегі оқу бағдарламаларымен
үйлестірілген бағдарламаға қатысуға арқызданады. Әлібек бұл оқу корпусының үлкенің жиын
сағасынан немесе тағы да жиындардан жақсырақ, жаңадан оқуға азайып бара жатқанын сезіне білді.

Кешке, университет аумағындағы кештеңге қатысып, танысқануға тырылды. Ол қызығушылығымен
танысқады, бірақ жаңа ортада айналасындағы бақытты жандарды түсінуді немесе танысуды қиын тапты.
<end_of_turn>

=== Fine-Tuned Adapter Answer ===
Айман есімді студент алғаш рет Назарбаев Университетіне келді.Ол өз заманын еңбекқор және осы
мүмкіндікті танымал университет үшін пайдалануға дайын.

Күн сайын таңертең сағатқа қарсы, Айман таңертең қолымды жейде киіп, оның ноутбугін
қабылдайды.Ол университет жанындағы кафеде сағат 9-да азық-түлік кешкі ас ішіп қалады және
бірнеше минуттан кейін сыныпқа қайтып келеді.Келген сағат 11-ге дейін ол дәрістер, семинарлар
мен іс-шараларға қатысады.Кеш, Айман баяу үйге баратын жолда жергілікті кафеде келеді, өзіне
көңілді және тәтті демі немесе сүт пен дәнді пісіруді алғаш рет иеленеді.Кешірімге келіп, Айман
өз үйге барады, мұғалімдер мен студенттермен бірге болатын жаңа махаббатты және қиын сыныптарын
есте сақтайды.Оның академиялық әл-ауқаты үшін айтарлықтай күш-жігерден кейін, ол әрқашан оның
барлығына ризашылық білдіретін әдемі демалу күн сайын бауырластармен бірге болатын кенеттен
жұмыстан шығылған.<end_of_turn>
```

Importantly, it does not tend to generate words/symbols from other languages unlike the base model as can be seen on the record below, yet both outputs are not straight to the points since the morphological structure is trickier here though our model still sticks to the only Kazakh-text generation:

```
User prompt: Translate the following technical paragraph into Kazakh: Deep learning models rely on multiple layers of abstraction to learn hierarchical representations from data.

=== Base Gemma-3 Answer ===
Here are a few options for translating the paragraph "Deep learning models rely on multiple layers of abstraction to learn hierarchical representations from data" into Kazakh, with slightly different
nuances. Here's a breakdown and the most recommended options:

**Option 1 (Most Precise & Technical - Recommended):**

"Дәппелі үйрентуіш моделі негізінде деректерден жеңерлік қабаттанған білулерді игеру үшін бірнеше абстракция қабаттары екендігі сақталады."

* **Дәппелі үйрентуіш моделі (Dappeli ÿërentuış modeli):** Deep learning models
* **негізінде (neğiziñde):** based on, relying on
* **деректерден (derekterden):** from data
* **жеңерлік қабаттанған білулерді (žeñerlik qabattançan

=== Fine-Tuned Adapter Answer ===
Жасушалық оқыту моделі деректерді үлкен және әр түрлі түрде алу үшін көп қабаттардағы қабаттарды қолданады.Әр қабат бірегей қателіктер мен функцияларды түсінуге үйретіліп, әрбір әрекетті ұйымдастыруға
үйретілген.Бұл әр қабаттан өту деректердің құрылымын түсінуге көмектеседі және одан әрі, күрделі әлемнің жақсы өкілдігі жасауға көмектеседі.<end_of_turn>
```

To sum up, the nuanced Kazakh language generation tasks, such as summarization, grammar correction, redundancy check, instruction following, language dexterity and creativity are more pronounced within our fine-tuned model, which is a testament to QLoRA adapters and specific token-rich Kazakh datasets.

7. Conclusion and Possible Future Work (5%)

The initiative of this project represents an important advancement in minimizing the gap in Kazakh-language NLP by launching the first dedicated language model designed explicitly for Kazakh. According to the evaluation results, the improved QLoRA model demonstrates significantly improved performance compared to the basic Gemma-3 4B model in many aspects of Kazakh language processing. The evaluation performance remained stable throughout the training, which indicates the stability of the model and the effectiveness of defining control points. The model perfectly copes with complex tasks, such as grammar correction, generalization, and stylistic refinement, where it consistently produces more accurate, concise, and instructional results compared to the original version. It is noteworthy that this also demonstrates stricter adherence to Kazakh language rules and language structures, avoiding non-Kazakh lexemes and excessive verbosity. These results confirm the effectiveness of our approach combining LoRa adapters with a carefully selected dataset in Kazakh, rich in lexemes.

Even with the significant advancements made, some restrictions are still present. The variety and size of the training dataset, although adequate for preliminary modeling, could be enhanced to more effectively reflect the entire linguistic range of the Kazakh language. Moreover, restricted access to advanced computing resources limited our capacity to test larger model structures and extended training periods, possibly enhancing model accuracy and generalization for further model development cycles.

In future work, we can expand the data set both in size and diversity, paying more attention to improving the quality of translations in order to better reflect the nuances of the Kazakh language. In addition, we would like to experiment with larger model architectures, such as versions with parameters 12B and 27B, in order to study their ability and operability to capture complex linguistic patterns. Future experiments may also include systematic tuning of hyperparameters to optimize learning stability and output quality. It is expected that these areas will further enhance the effectiveness and generalizability of the model in the tasks of Kazakh NLP.

8. References (5%)

Anastasopoulos, A., & Neubig, G. (2019). Pushing the limits of low-resource morphological inflection. *Proceedings of the Association for Computational Linguistics (ACL)*.

Blasi, D. E., Anastasopoulos, A., & Neubig, G. (2021). Systematic inequalities in language technology performance across the world's languages. *arXiv preprint* arXiv:2110.06733. https://arxiv.org/abs/2110.06733

Blasi, D. E., Anastasopoulos, A., & Neubig, G. (2022). Systematic inequalities in language technology performance across the world’s languages. *Proceedings of ACL 2022*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*. https://arxiv.org/abs/2106.09685

Joshi, P., Santy, S., Budhiraja, A., Bali, K., & Choudhury, M. (2020). The state and fate of linguistic diversity and inclusion in the NLP world. *Proceedings of ACL 2020*.

Köksal, A., Thaler, M., Imani, A., Üstün, A., Korhonen, A., & Schütze, H. (2024). MURI: High-quality instruction tuning datasets for low-resource languages via reverse instructions. *arXiv preprint* arXiv:2409.12958. https://arxiv.org/abs/2409.12958

Koto, F., Joshi, R., Mukhituly, N., Wang, Y., Xie, Z., Pal, R., Orel, D., Mullah, P., Turmakhan, D., Goloburda, M., Kamran, M., Ghosh, S., Jia, B., Mansurov, J., Togmanov, M., Banerjee, D., Laiyk, N., Sakip, A., Han, X., Kochmar, E., Aji, A. F., Singh, A., Jadhav, A. A., Katipomu, S., Kamboj, S., Choudhury, M., Gosal, G., Ramakrishnan, G., Mishra, B., Chandran, S., Sheinin, A., Vassilieva, N., Sengupta, N., Murray, L., & Nakov, P. (2025). Llama-3.1-Sherkala-8B-Chat: An open large language model for Kazakh. *arXiv preprint* arXiv:2503.01493. https://arxiv.org/abs/2503.01493

Kudugunta, S., Caswell, I., Zhang, B., Garcia, X., Choquette-Choo, C. A., Lee, K., Xin, D., Kusupati, A., Stella, R., Bapna, A., & Firat, O. (2023). MADLAD-400: A multilingual and document-level large audited dataset. *arXiv preprint* arXiv:2309.04662. https://doi.org/10.48550/arXiv.2309.04662

Laiyk, N., Orel, D., Joshi, R., Goloburda, M., Wang, Y., Nakov, P., & Koto, F. (2025). Instruction tuning on public government and cultural data for low-resource language: A case study in Kazakh. *arXiv preprint* arXiv:2502.13647. https://arxiv.org/abs/2502.13647

Li, C., Yang, W., Zhang, J., Lu, J., Wang, S., & Zong, C. (2024). X-Instruction: Aligning language model in low-resource languages with self-curated cross-lingual instructions. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Findings of the Association for Computational Linguistics: ACL 2024* (pp. 546–566). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.30

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Nguyen, T., Nguyen, C. V., Lai, V. D., Man, H., Ngo, N. T., Dernoncourt, F., Rossi, R. A., & Nguyen, T. H. (2023). CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *arXiv preprint* arXiv:2309.09400. https://doi.org/10.48550/arXiv.2309.09400

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: A method for automatic evaluation of machine translation. *Proceedings of ACL 2002*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, *21*(140), 1–67.

Yeshpanov, R., Polonskaya, A., & Varol, H. A. (2024, May). *KazParC: Kazakh Parallel Corpus for Machine Translation*. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 9633–9644). Torino, Italia: ELRA and ICCL. https://aclanthology.org/2024.lrec-main.842/

Yeshpanov, R., Efimov, P., Boytsov, L., Shalkarbayuli, A., & Braslavski, P. (2024, May). *KazQAD: Kazakh Open-Domain Question Answering Dataset*. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 9645–9656). Torino, Italia: ELRA and ICCL. https://aclanthology.org/2024.lrec-main.843/

Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., & Ma, Y. (2024). Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.