Emotion detection in russian language

Sofia Samoylova, Dmitrii Sutyi, Evgenii Kallimulin

June 2025

Abstract

The growth of user-generated content across global platforms has made multilingual emotion detection a critical area of research. This paper presents our system and results for the task inspired by SemEval-2025 Task 11: Bridging the Gap in Text-Based Emotion Detection[1]. Our work concentrates on Track A, which addresses multi-label emotion detection, with a specific focus on improving performance for the Russian language. To tackle this challenge, we use a large language model pre-trained on Russian texts, which we further adapt for multi-label classification. We employ a Parameter-Efficient Fine-Tuning (PEFT) approach, specifically Low-Rank Adaptation (LoRA), to effectively train the model on the target task. Our model demonstrates state-of-the-art performance, securing the 1st place rank for the Russian language on the official evaluation metric. This achievement highlights the effectiveness of our model choice and fine-tuning strategy for capturing complex emotional expressions in a specific linguistic context. The code for our project is available at: `https://github.com/sofismv/emotion-detection`.

1 Introduction

The rapid spread of social media and digital communication platforms has transformed how users express themselves, generating vast amounts of text rich with emotional content. Analyzing these emotions at scale is a critical task in Natural Language Processing (NLP), with significant applications ranging from market intelligence and customer feedback analysis to mental health monitoring and social science research (Nandwani and Verma, 2021) [1]. However, understanding human emotion from text is inherently complex; a single statement can simultaneously convey multiple emotions, making multi-label emotion detection a particularly challenging yet vital area of study.

While most research has focused on creating universally applicable multilingual models, a parallel and equally important challenge is the ability to achieve state-of-the-art performance for specific, high-resource, yet structurally complex languages. General-purpose multilingual models, while broad in scope, may not capture the unique nuances, cultural context, and idiomatic expressions of a

[1] `https://github.com/emotion-analysis-project/SemEval2025-Task11`

single language as effectively as a more targeted approach. Our work addresses this gap, focusing specifically on advancing the state-of-the-art for multi-label emotion detection in the Russian language.

This paper presents our system and results for SemEval-2025 Task 11: "Bridging the Gap in Text-Based Emotion Detection" (Muhammad et al., 2025) [2]. Our participation was limited to Track A: Multi-label Emotion Detection, where our primary objective was to develop a high-performing model for the Russian subset of the dataset. Also we adapted our solution for Track B: Emotion intensity.

Our approach is based on the strategic fine-tuning of a large-scale pre-trained language model selected for its deep understanding of the Russian language. Instead of conventional full-model fine-tuning, which is computationally expensive and can lead to catastrophic forgetting, we employed a Parameter-Efficient Fine-Tuning (PEFT) methodology, specifically Low-Rank Adaptation (LoRA) (Hu et al., 2021)[3]. This technique allows highly efficient adaptation by training only a small number of additional parameters, preserving the model's extensive pre-trained knowledge while precisely tuning it for the downstream task.

Our focused strategy proved to be successful. In the official competition results our system secured the 1st place rank for Track A and the 2nd for Track B for Russian language, outperforming all other submissions on the official evaluation metric. This result validates our core hypothesis: that a focused, parameter-efficient adaptation of a strong base model is a superior strategy for achieving state-of-the-art performance on a specific language compared to more generalized, general multilingual approaches. This paper details our methodology, experimental setup, and the results that led to this achievement.

1.1 Team

- **Sofia Samoylova** field research, provided experiments and code.
- **Evgenii Kallimulin** provided experiments and paper preparing.
- **Dmitrii Sutyi** field research and paper preparing.

2 Related Work

The task of automatically detecting emotions in text is a long-standing and fundamental area within Natural Language Processing (NLP), which has evolved significantly with the advent of modern machine learning techniques. This section outlines the primary paradigms of emotion detection, situating our work within the current state-of-the-art.

The advent of deep learning, and particularly the Transformer architecture (Vaswani et al., 2017)[4], marked a paradigm shift. Pre-trained models like BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019)[5] became the de-facto standard for a wide range of text classification tasks. The common approach involves fine-tuning a pre-trained encoder model by adding a classification head on top of a pooled representation, such as the [CLS] token's output. Our BERT-based baseline leverages RuBERT, a model specifically adapted for Russian downstream tasks by Kuratov et al. (2019) [6]. The development of RuBERT involved initializing from the original BERT architecture and its English pre-trained weights, and then continuing the self-supervised pre-training phase on a massive Russian-language corpus (including Wikipedia and news data). This process gave the model a deep understanding of Russian vocabulary, syntax, and semantics, making it a powerful starting point for task-specific fine-tuning.

More recently, the landscape has been reshaped by the emergence of Large Language Models (LLMs) with billions of parameters, such as the GPT family (Brown et al., 2020)[7]. These models have demonstrated remarkable capabilities in performing tasks with little to no task-specific training through prompting. This has given rise to two main strategies: zero-shot prompting, where the model is provided only with a natural language instruction, and few-shot in-context learning, where a few examples of the task are included in the prompt to guide the model's output. While these prompting methods offer great flexibility and remove the need for fine-tuning, their performance can be inconsistent and often lags behind that of fully fine-tuned models on specific, well-defined tasks.

To overcome the prohibitive computational costs of fully fine-tuning LLMs, Parameter-Efficient Fine-Tuning (PEFT) methods have gained prominence. Among these, Low-Rank Adaptation (LoRA) (Hu et al., 2021)[**lorapaper**] has become particularly popular. LoRA freezes the pre-trained model weights and injects trainable, low-rank matrices into the Transformer layers, drastically reducing the number of trainable parameters and memory requirements. A further advancement, Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023)[8], combines LoRA with 4-bit quantization of the base model's weights. This technique further democratizes the fine-tuning of massive models, making it possible to adapt models with billions of parameters on consumer-grade or freely available hardware.

3 Model Description

3.1 Fine-tuned BERT-based Classifier

Our first baseline employs a traditional and robust fine-tuning methodology using an encoder-only Transformer architecture.

We selected DeepPavlov/rubert-base-cased-conversational, a BERT-based model pre-trained specifically on Russian conversational text, making it a strong candidate for this task. The model was loaded using the AutoModelForSequenceClassification class from the Hugging Face Transformers library, configured for multi-label classification (problem_type="multi_label_classification"). This architecture (fig. 1) adds a single linear layer (a classification head) on top of the final hidden state of the [CLS] token.

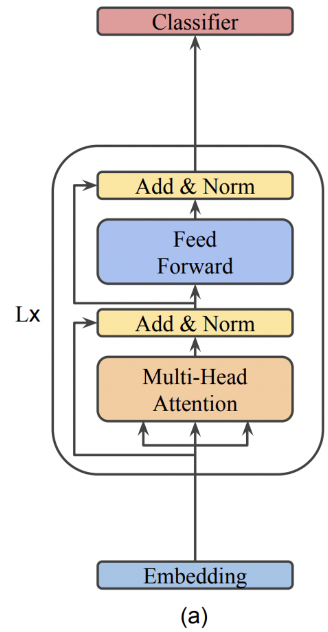

Figure 1: Bert architecture.[9]

The fine-tuning process updated all weights of the base model and the newly added classification head. To handle the multi-label nature of the task, a sigmoid function was applied to the output logits of each emotion class, and the model was trained using a Binary Cross-Entropy loss. Following training, we further optimized the classification threshold (from 0.1 to 0.9) on the validation set to maximize the final macro F1-score.

3.2 Zero-Shot Prompting with an LLM

This approach explores the natural capabilities of a large language model to perform emotion detection without any task-specific training, relying solely on prompt engineering.

We utilized yandex/YandexGPT-5-Lite-8B-instruct, an 8-billion parameter instruction-tuned decoder-only LLM. The model was loaded in 4-bit precision using BitsAndBytesConfig to manage memory usage. For each text instance in the test set, we constructed a detailed prompt in Russian. This prompt assigned the model the role of an "emotion analysis expert," provided a fixed list of possible emotions, and laid out strict rules for classification (e.g., to only identify explicitly stated emotions).

The model's task was purely generative: it received the prompt containing the text and generated a response listing the detected emotions. A postprocessing function was then required to parse this free-text response (e.g., "[гнев, удивление]") and convert it into a binary vector corresponding to our six emotion classes for metric calculation. No gradient updates were performed on the model.

3.3 Few-Shot Prompting with an LLM

This method extends the zero-shot approach by incorporating in-context learning, providing the model with relevant examples to guide its prediction.

This approach uses the same yandex/YandexGPT-5-Lite-8B-instruct model as in the zero-shot baseline. However, the prompt construction is dynamic and retrieval-augmented. First, we pre-computed vector embeddings for every text in the training set using a sentence-transformer model (paraphrase-multilingual-MiniLM-L12-v2)[2].

For each new text instance from the test set, we performed a semantic search: we embedded the test text and used cosine similarity to find the k=3 most similar examples from the training data. These retrieved examples (text and their corresponding ground-truth emotion labels) were then dynamically inserted into the prompt. This provides the LLM with relevant, in-context demonstrations of the task immediately before it is asked to analyze the new text. The generation and output parsing steps remained identical to the zero-shot approach. This strategy tests the model's ability to learn from examples provided directly within the context window.

3.4 Fine-tuned LLM

Our main approach is fine-tuning russian-based LLM (YandexGPT-5-Lite-8B-instruct) for classification.

We take the base decoder model, pass our tokenized input into the decoder and get contextualized embeddings (hidden_states) from the last layer of decoder. We then pass those hidden states through a linear layer to get the logits for each of the input tokens. We find the last non-padding token in the input sequence and take the logits for it and calculate the final cross-entropy loss. The architecture is presented in Fig. 2.

For Track B (emotion intensity classification), we reformulate the task as a single-label, multi-class classification over text-emotion pairs. Each input is constructed by concatenating the original text with a candidate emotion (e.g., "Text: <text> Emotion: joy"), and the full string is tokenized and passed through the LLM. For each original text six pairs are generated — one per emotion. The model is trained to predict the intensity level of the given emotion on a discrete scale from 0 to 3 (no emotion to high intensity).

[2] `https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`

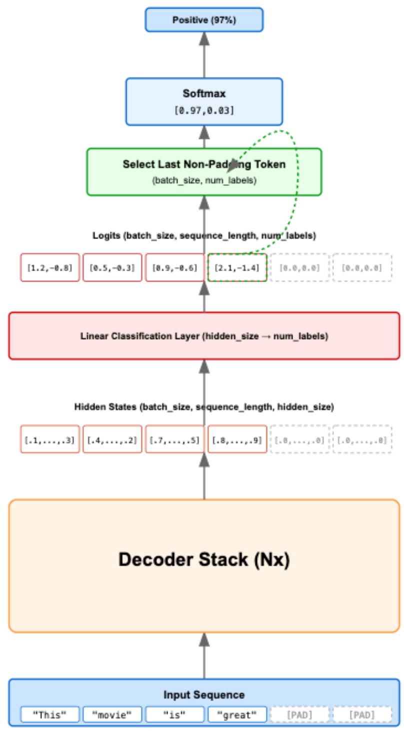

Figure 2: LLM classification architecture.[10]

4 Dataset

To train and evaluate our methods, we use the official dataset BRIGHTER-emotion-categories[3], which includes a Russian-language subset and was provided by the SemEval Task 11 organization. Each entry corresponds to a single sentence or short text written in social media. These texts are annotated with binary indicators for six basic emotions: *anger*, *disgust*, *fear*, *joy*, *sadness*, and *surprise*.

This is a sample instance from the dataset:

> "Я ненавижу свою бессонницу !!! Не надо бы было эти 2 недели засыпать в 3 ночи((теперь привычка((((((((("
> Labels: anger, sadness

On Tab. 1 you can see the statistics for the Russian subset of the BRIGHTER

[3] `https://huggingface.co/datasets/brighter-dataset/BRIGHTER-emotion-categories`

Emotion Categories dataset [2]. The dataset is divided into 3 parts: training, validation (dev), and test subsets. Each text sample is annotated with a multi-label representation of six categorical emotions.

	Russian Subset
Train size	2,679
Validation size	398
Test size	2,000
Total size	5,077
Emotion labels	`{anger, disgust, fear, joy, sadness, surprise}`
Task type	Multi-label classification
Language ISO Code	`ru`

Table 1: Statistics of the Russian portion of the BRIGHTER Emotion Categories dataset. Each example contains binary annotations for six basic emotions.

This dataset was not collected manually for this project but rather reused from publicly released resources. The BRIGHTER dataset is designed to address the lack of human-annotated emotion datasets for low-resource languages. Texts were sourced from multiple domains and manually annotated for emotion presence.

For track B (Emotion Intensity) we utilize the BRIGHTER Emotion Intensities dataset[4], which is derived from the same BRIGHTER initiative [2]. Each example is annotated with an intensity score for six basic emotions.

Table 2 shows the statistics for the Russian configuration of this dataset.

	Russian Subset
Train examples	2,220
Validation examples	343
Test examples	650
Total examples	3,213
Emotion annotations	Intensity scores (integer 0-3) for six emotions
Task type	Regression / fine-grained classification
Language ISO Code	`ru`

Table 2: Statistics of the Russian portion of the BRIGHTER Emotion Intensities dataset. Each example contains numerical scores for six emotion categories.

This is a sample instance from the dataset:

[4] `https://huggingface.co/datasets/brighter-dataset/BRIGHTER-emotion-intensities`

"Зачет на 10!!! Даааа!"
Labels: joy 2

Unlike Track A, this dataset enables more nuanced modeling and evaluation of emotional expression in text. It is suitable for training models that estimate the *degree* of emotional intensity rather than merely detecting presence or absence.

5 Experiments

5.1 Metrics

To evaluate our approach, we used the macro-averaged F1 score (**F1-macro**), which is the official evaluation metric for Track A: Multilabel Emotion Detection. In addition to the F1 score, we also report *precision* and *recall* to provide a more complete picture of the model's performance.

In the multilabel setting, each example may be assigned multiple labels. Therefore, F1-macro is computed by first calculating the F1 score for each emotion category individually, and then averaging them:

$$\mathrm{F1}_{\mathrm{macro}} = \frac{1}{L} \sum_{i=1}^{L} \frac{2 \cdot \mathrm{Precision}_i \cdot \mathrm{Recall}_i}{\mathrm{Precision}_i + \mathrm{Recall}_i} \tag{1}$$

where L is the number of labels (in our case, six emotions), and $\mathrm{Precision}_i$ and Recall_i are defined for each label i as:

$$\mathrm{Precision}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i} \tag{2}$$

$$\mathrm{Recall}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i} \tag{3}$$

Here, TP_i, FP_i, and FN_i denote the number of true positives, false positives, and false negatives for label i, respectively.

Macro-averaging ensures that each emotion category contributes equally to the overall score, regardless of its frequency in the dataset.

For Track B: Emotion Intensity, we evaluate our model using the Pearson correlation coefficient, which is the official evaluation metric for Track B, which measures the linear relationship between the predicted emotion intensities and the gold (annotated) intensities. This metric is particularly suited for regression-style tasks or ordinal classification settings, where the relative order and distance between predicted values are important.

Given a set of n gold labels $\{y_1, y_2, \ldots, y_n\}$ and corresponding predictions $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$, the Pearson correlation coefficient r is computed as:

$$r = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2} \cdot \sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{4}$$

where $\bar{y}$ and $\bar{\hat{y}}$ are the means of the true and predicted intensity scores, respectively.

Although our model formulates the task as a 4-class classification problem (with discrete labels from 0 to 3), we treat the predicted class labels as numerical values to compute the Pearson correlation. This allows us to capture how well the predicted intensity levels align with the true emotion intensities, even in the absence of probabilistic or continuous outputs.

A higher Pearson correlation indicates that the model better preserves the ordinal structure and relative intensity of emotions in its predictions.

5.2 Experiment Setup

We used the official data splits provided in the BRIGHTER Emotion Categories dataset, consisting of separate training, development, and test sets for each language. All models were trained and evaluated on the Russian subset of the dataset.

For fine-tuning experiments, we performed hyperparameter optimization using the Optuna framework. The seed was fixed to 42 for reproducibility. Evaluation during training was based on the macro F1-score computed only on the development set. For our calculation, we used NVIDIA GeForce RTX 2080 Ti.

For LLM based methods we used YandexGPT-5-Lite-8B-instruct [11] and employed QLoRA. This approach significantly reduces the GPU memory requirements, enabling fine-tuning of 8-billion parameter LLMs accessible on free Google Colab environments. This setup requires approximately 8GB of GPU memory during training with this setup.

5.3 Baselines

We experimented with multiple approaches before finalizing our chosen architecture. Below, we describe each of the explored methods and justify our final design choice based on empirical performance and architectural advantages.

5.3.1 BERT-based

We initially experimented with fine-tuning a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model. A classification head was added on top of the [CLS] token representation to predict the emotion class.

For conversional Russian language best model is DeepPavlov/rubert-base-cased-conversational.

We performed hyperparameter tuning with Optuna over the following search space:

- Learning rate: $[1\text{e}{-5}, 5\text{e}{-4}]$

- Batch size: $\{8, 16, 32\}$
- Weight decay: $[0.0, 0.3]$
- Warmup steps: $[0, 500]$
- Number of epochs: $[2, 7]$

The model was trained using the Hugging Face `Trainer` API with sigmoid activation at the output and binary cross-entropy loss to handle the multilabel classification setup.

5.3.2 Zero-shot prompting LLM

To explore the potential of large language models (LLMs) in a prompt-based setup, we tried zero-shot prompting using Yandex-GPT model. We provided a direct instruction to detect emotions from list in text.

The prompt instructed the model to identify explicitly expressed emotions in the text from a predefined list: anger, disgust, fear, joy, sadness, surprise. It also specified that if no emotions are present, the output should be an empty list.

Example prompt:

> Ты эксперт по анализу эмоций в тексте.
> Определи, какие эмоции выражены в тексте из списка [гнев, отвращение, страх, радость, грусть, удивление.]
> Эмоций может быть несколько, а может и вовсе не быть. Формат вывода: только названия эмоций. Если нет эмоций, то оставь пустой список.
> ВАЖНЫЕ ПРАВИЛА:
> - выбирай эмоцию, ТОЛЬКО если она выражена ЯВНО через конкретные слова, фразы или контекст
> - НЕ додумывай скрытые эмоции - только то, что написано прямо
> - При сомнениях не выбирай эмоцию
> Пример вывода: [гнев]
> Проанализируй этот текст по этим критериям:
> Текст: {query text}
> Ответ:

Outputs were post-processed into binary label vectors via keyword matching.

The model performs well on clearly marked emotions but tends to miss subtler emotional signals.

5.3.3 Few-shot prompting LLM

Few-shot prompting extended zero-shot approach by including several labeled examples from train data with most similar embedding values to current text in the prompt before querying the model. We assume that conditioning LLM with relevant in-domain examples helps it to better capture the nuances of emotional signals.

Example prompt:

> Ты эксперт по анализу эмоций в тексте.
> Определи, какие эмоции выражены в тексте из списка [гнев, отвращение, страх, радость, грусть, удивление.]
> Эмоций может быть несколько, а может и вовсе не быть. Формат вывода: только названия эмоций. Если нет эмоций, то оставь пустой список.
> ВАЖНЫЕ ПРАВИЛА:
> - выбирай эмоцию, ТОЛЬКО если она выражена ЯВНО через конкретные слова, фразы или контекст
> - НЕ додумывай скрытые эмоции - только то, что написано прямо
> - При сомнениях не выбирай эмоцию
>
> Примеры:
> Текст: {example text}
> Ответ: {emotions}
>
> Проанализируй этот текст по этим критериям:
> Текст: {query text}
> Ответ:

5.3.4 Fine-Tuning LLM

For our fine-tuning experiments, we utilized YandexGPT-5-Lite-8B-instruct, a new and lightweight model. The adaptation employed:

Setup

- **Base Model**: YandexGPT-5-Lite-8B (8B parameters) with 4-bit NF quantization
- **Adapter**: LoRA ($r = 8$, $\alpha = 16$) applied to all linear layers + classifier head
- **Training Hardware**: Single NVIDIA GeForce RTX 2080 Ti
- **Training**:
  - Batch size: 4 (effective 16 via $\times 4$ gradient accumulation)
  - Cosine LR scheduler: 3×10^{-5} base rate with 5% warmup
  - 4-epoch training with FP16 mixed precision
  - Weight decay: 0.01, max grad norm: 1.0

- **Optimizations**:
  - Sequence length: 512 tokens (padded to multiples of 16)
  - Grouped by length for efficient batching
  - Optuna-tuned hyperparameters

Component	**Setting**
Quantization	NF4 + double quant
LoRA Rank (r)	8
Batch Size	4 (effective 16)
Base LR	3×10^{-5}
Warmup	5% of steps
Epochs	4

Table 3: Key training configurations

For track B setup was the same.

6 Results

Our method achieved **state-of-the-art performance** on the Russian subset of the BRIGHTER Emotion Categories dataset. Specifically, we outperformed the top-1 leaderboard[5] score in terms of macro F1-score on the test set. This demonstrates the effectiveness of our emotion classification approach compared to our baselines.

	F1	precision	recall
PA-oneteam-1	0.9087	-	-
Finetune LLM	**0.9093**	**0.9293**	**0.8913**
ruBERT	0.8734	0.8983	0.8502
Zero-shot	0.8281	0.9096	0.7794
Few-shot	0.8583	0.8804	0.8445

Table 4: Macro-averaged scores across all emotions. PA-oneteam-1 refers to the top-ranked team on the leaderboard.

Moreover, by leveraging QLoRA and 4-bit quantization, we significantly reduced the hardware requirements for model fine-tuning. The entire training process was conducted on a single GPU with less than 8GB of memory usage, making our solution highly accessible and cost-effective. Despite its low resource

[5] `https://docs.google.com/spreadsheets/d/1IgPfmL0z9Lc7GCVY3Lw0sHEGQTv0jtZ0/edit?usp=sharing&ouid=111873688488571848655&rtpof=true&sd=true`

requirements, our approach achieved superior results, demonstrating that large-scale instruction-tuned LLMs can be effectively adapted to specific tasks with minimal computational infrastructure.

Per-class metrics (Table 5) show strong performance across all six emotion categories, with the highest F1-scores for *Joy* (0.9406) and *Fear* (0.9340). Even traditionally challenging categories like *Disgust* (0.9004) and *Surprise* (0.8880) show robust results, indicating that the model captures both frequent and rare emotion expressions effectively.

	F1	precision	recall
Anger	0.9116	0.9349	0.8894
Fear	0.9340	0.9519	0.9167
Joy	0.9406	0.9381	0.9430
Disgust	0.9004	0.9541	0.8525
Sadness	0.8815	0.9225	0.8440
Surprise	0.8880	0.8740	0.9024

Table 5: Per-class performance metrics (F1, precision, and recall) for each emotion category.

We also measured training and evaluation performance in terms of runtime and throughput. On the selected configuration using QLoRA with 4-bit quantization, one training epoch took approximately 15 minutes on NVIDIA Geforce RTX 3060(2679 samples in train) and take 8GB GPU memory. During inference, the throughput reached 8.72 samples per second and take 6GB GPU memory, confirming the efficiency of the method even on limited hardware.

Table 6 presents the performance of our model on the test set for Track B (Emotion Intensity) on the Russian language subset. As shown, the model achieves consistently high Pearson correlation scores across all six emotion categories, with an overall average score of **0.9209**.

Emotion	**Pearson Correlation**
Anger	0.9527
Disgust	0.9299
Fear	0.9395
Joy	0.9035
Sadness	0.8946
Surprise	0.9017
Average	**0.9209**

Table 6: Pearson correlation scores for each emotion in Track B (Russian subset).

Our method achieved **2nd place** on the Russian subset of the BRIGHTER

Emotion intensities dataset. Leaderboard[6]

7 Conclusion

In this paper, we presented our system for the Russian language track of SemEval-2025 Task 11: Bridging the Gap in Text-Based Emotion Detection. Our primary objective was to develop a state-of-the-art model for multi-label emotion classification, moving beyond generalized multilingual approaches to create a highly specialized and effective system for a specific linguistic context.

To achieve this, we conducted a comparative analysis of four distinct methodologies. Our proposed system, based on the YandexGPT-5-Lite-8B-instruct model, secured 1st place on both the official competition leaderboard and the final post-challenge ranking, demonstrating the robustness and consistency of our approach. A crucial factor in our success was the selection of a base model with a strong foundational understanding of the Russian language. We then employed Quantized Low-Rank Adaptation (QLoRA) to efficiently adapt this powerful base model to the specific task. This strategy allowed us to achieve these top-tier results without access to large-scale computational clusters, conducting the entire fine-tuning process on a single consumer-grade GPU.

Our experiments demonstrate a clear performance hierarchy, with the fine-tuned LLM achieving a final macro F1-score of 0.9093, significantly outperforming all baselines. This result validates our central hypothesis: the combination of a powerful, language-specific base model with a targeted, parameter-efficient adaptation technique is a superior strategy for achieving state-of-the-art performance. What makes this achievement particularly significant is that these results were obtained with remarkable computational efficiency, proving that a focused approach can outperform more generalized methods without requiring extensive resources.

References

[1] Pansy Nandwani and Rupali Verma. *A review on sentiment analysis and emotion detection from text.* 2021. URL: https://www.researchgate.net/publication/354196437_A_review_on_sentiment_analysis_and_emotion_detection_from_text.

[2] Shamsuddeen Hassan Muhammad et al. *BRIGHTER: BRIdging the Gap in Human-Annotated Textual Emotion Recognition Datasets for 28 Languages.* 2025. URL: https://arxiv.org/abs/2502.11926.

[3] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models.* 2021. arXiv: 2106.09685 [cs.CL]. URL: https://arxiv.org/abs/2106.09685.

[6] https://docs.google.com/spreadsheets/d/1LQ552iLJKaXmZihPvAFvRpDapDN9RDvg/edit?gid=1994287181#gid=1994287181

[4] Ashish Vaswani et al. *Attention is all you need*. 2017. arXiv: 1706.03762. URL: https://arxiv.org/pdf/1706.03762.

[5] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805. URL: https://arxiv.org/pdf/1810.04805.

[6] Yuri Kuratov and Mikhail Arkhipov. *Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language*. 2019. arXiv: 1905.07213. URL: https://arxiv.org/abs/1905.07213.

[7] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165. URL: https://arxiv.org/pdf/2005.14165.

[8] Tim Dettmers et al. *QLoRA: Efficient Finetuning of Quantized LLMs*. 2023. arXiv: 2305.14314. URL: https://arxiv.org/abs/2305.14314.

[9] Zhiqing Sun et al. *MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices*. 2020. arXiv: 2004.02984v2. URL: https://arxiv.org/pdf/2004.02984v2.

[10] Atharva Ingle. *LLMs are machine learning classifiers*. URL: https://wandb.ai/gladiator/LLMs-as-classifiers/reports/LLMs-are-machine-learning-classifiers--VmlldzoxMTEwNzUyNA.

[11] Yandex. *YandexGPT-5-Lite-8B-Instruct*. https://huggingface.co/yandex/YandexGPT-5-Lite-8B-instruct. 2025.