Ministry of Science and Higher Educationof the
Russian Federation
ITMO University

Faculty of Digital Transformation
Educational program: Big Data and Machine Learning

REPORT

on research internship

Task topic: Benchmark for assessing the quality of language models when working with the Russian language

Student: Abdurakhimov Muslimbek Abdulboqi ogli, J1433C

Head of Practice from ITMO University: Khodorchenko Maria Andreevna, Senior Researcher at research center "Strong artificial intelligence in industry"

Practice completed with grade _____

Commission member signatures:

_____
fullname

(signature)

_____
fullname

(signature)

_____
fullname

(signature)

Date ____

St. Petersburg

2024

ESSAY

Abdurakhimov M.A. Benchmark for assessing the quality of language models when working with the Russian language

Report contains 30 pages, 16 figures, 2 tables, 24 sources.

Keywords: Language Models, Evaluation Metrics, Model evaluation, natural language understanding, benchmarks, NLP models, language modelling, general language understanding evaluation.

By focusing specifically on language models working with the Russian language, the report addresses the need for tailored evaluation methods and benchmarks to assess their quality effectively. As the field of natural language processing continues to evolve rapidly, particularly in multilingual settings, it becomes increasingly important to develop robust evaluation frameworks tailored to specific languages and linguistic contexts.

Through a combination of theoretical insights, empirical analysis, and practical considerations, the report offers valuable contributions to the field of language model evaluation. By examining various evaluation metrics and methodologies, the report provides insights into the strengths and limitations of existing approaches, as well as opportunities for future research and development.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Background and motivation

Prompting, a method where specific text inputs guide machine learning models, has gained popularity recently [15]. This approach offers control over model outputs by feeding them particular prompts. Our interest lies in improving the assessment of language models, assuming that good models should handle prompt variations consistently. We propose a stability coefficient to better evaluate and interpret language model results based on their stability to prompt changes.

1.2. Objective and scope

In recent years, prompting as a directed interaction methodology with machine learning models has gained increasing popularity. This method provides the ability to control the output of models by providing them with specific textual input data or prompts. In this work, our main goal is to improve the assessment of language models, based on the assumption that a good model should be stable to the variability of prompts. In our work, we propose a stability coefficient that allows for more effective evaluation and interpretation of the results of language models depending on their stability to changes in the input prompt.

In the context of evaluating language models and their response to rapid changes, it is important to establish reliable evaluation metrics that accurately reflect the performance and stability of the models. Traditional evaluation metrics in natural language processing tasks, such as BLEU scores and classification metrics, provide valuable information about the model's performance but may not fully reflect nuances related to prompt variations. One of the key issues to address is considering the stability and reliability of models in the face of changing input data or prompts. Currently, there is no standard approach to account for model stability in response to changes in input prompts, limiting our ability to meaningfully evaluate and compare models.

The stability coefficient can be calculated by measuring the similarity or consistency of model results when presented with different prompts for the same task. In particular, this may involve calculating cosine similarity or other similarity

metrics between the output data generated by the model in response to different prompts. A higher stability coefficient indicates that the model provides more consistent responses regardless of prompt changes, reflecting its stability in processing various input conditions.

2. LITERATURE REVIEW

2.1. Traditional evaluation metrics in NLP

In this section, we provide a comprehensive overview of the evaluation metrics employed in our study to assess the performance of language models on various benchmarks. Evaluation metrics are essential tools for quantifying the effectiveness of models across diverse linguistic tasks. We consider a range of metrics to ensure a thorough examination of the models' capabilities. Choice of evaluation metrics is highly task dependent.

Accuracy, Recall, Precision, F1 Score

Accuracy, as a performance measure, is straightforward and intuitive, representing the ratio of correctly predicted observations to the total number of observations. The common perception is that high accuracy implies an excellent model. While accuracy is indeed a valuable metric, its effectiveness is most pronounced in datasets where false positive and false negative values are approximately equal, creating a symmetrical distribution [8].

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ Predictions}$$

Recall (Sensitivity) reflects the completeness of a model, assessing its capability to identify all relevant instances (True Positive Rate). When the priority is minimizing False Negatives, Recall becomes crucial. A recall of 1.0 signifies the model produces no false negatives [7].

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Precision measures the exactness of a model, emphasizing its ability to return only relevant instances. In scenarios where minimizing False Positives is critical, Precision becomes a key metric. A perfect precision of 1.0 indicates the model

produces no false positives [7].

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

The F1 score serves as the harmonic mean of precision and recall, offering a balanced metric when there's an uneven class distribution. It is particularly useful in scenarios where achieving a balance between precision and recall is essential. The F1 score is defined as:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In summary, precision is favored when avoiding false positives is critical, recall is crucial when minimizing false negatives is essential, F1 score is suitable for balanced evaluation, and accuracy is appropriate when the class distribution is even.

BLEU:

BLEU (Bilingual Evaluation Understudy) is a cost-effective and swift metric to compute, demonstrating a strong correlation with human evaluation and showcasing language independence. It represented on a scale from 0 to 1, but often expressed as percentage. The higher the percentage, the better the result. This is the gradient used for interpretability scale for the BLEU evaluation metrics.

Figure 1. BLEU interpretability scale [17]

The score less than 10 % is almost useless to understand, while mode then 60 % is often better than human. The basic BLEU metric formula can be expressed as following:

$$\text{BLEU} = P_B * \exp\left(\sum_{n=0}^{n} W_n * \log * p_n\right)$$

Where pn is an n-gram precision that uses n-grams up to length N and positive weights wn that sum to one, and PB is the brevity penalty [17]. BLEU is widely used, but it has several limitations:

Lack of Consideration for Meaning: BLEU primarily relies on n-gram

overlap, which doesn't directly capture the semantic or contextual meaning of translated text [17].

Limited Consideration for Sentence Structure: The metric doesn't directly account for the overall structure of sentences.

Challenge with Morphologically Rich Languages: Morphologically rich languages, where words can have various forms and inflections, pose a challenge for BLEU. The metric may struggle to appropriately evaluate translations in languages with complex morphological structures [13].

Inability to handle synonyms: Synonyms or alternative expressions that convey similar meanings may not be adequately considered.

Overall. BLEU is valuable for its simplicity and efficiency, but its limitations make it less suitable for capturing the nuanced aspects of translation quality.

ROUGE:

ROUGE (Recall – Oriented Understudy for Gisting Evaluation) is a set of metrics used for the automatic evaluation of machine- generated text, particularly in the context summarization and document summarization. This metric comprises several variants, including ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. ROUGE-N, akin to BLEU-N, assesses the matching of n-grams between the machine-generated hypothesis and human-produced reference summaries [11].

METEOR:

The METEOR (Metric for Evaluation of Translation with Explicit ORdering) metric is an automatic evaluation metric used to assess the quality of machine-generated translations. The basic formula of the METEOR given by Lavie and Agarwal (2007) [18] can be found like this:

$$METEOR = \frac{(1-\beta) * P * R}{R + \beta * P} * (1 - \Upsilon)$$

Where, P is precision, R is Recall, β is a parameter that balances precision and recall, and Υ is a parameter that penalizes word order errors. The actual computation involves additional steps, such as stemming, synonym matching, and considering different levels of n-grams [6]. METEOR excels in evaluating machine-generated

translations, offering a thorough assessment. Its strengths lie in a holistic approach that considers unigram precision, recall, stemming, synonymy, and word order [6][9]. But sensitivity to parameter values could result in score variations, and may pose practical challenges.

2.2. Prompt-based Interaction with Language models

Prompt-based interaction with language models refers to the technique of guiding the outputs of machine learning models by providing specific textual inputs or prompts. This method has become increasingly popular as it allows users to control and fine-tune the behavior of language models for a wide range of tasks without the need for extensive retraining [20]. In prompt-based interaction, a prompt is a piece of text that is fed into a language model to elicit a desired response. The prompt can range from simple queries to complex instructions or contextual information. The structure and content of the prompt can significantly influence the quality and relevance of the model's output [18].

2.3. Model Explainibility Tools

Model explainability tools are essential for understanding, interpreting, and debugging the behavior of machine learning models, especially complex ones like language models [23]. These tools help to shed light on the decision-making processes of models, making them more transparent and trustworthy.

Captum is a PyTorch library for model interpretability, which includes various interpretability algorithms like Integrated Gradients, DeepLIFT, SHAP, and more. It provides a unified API to implement and visualize these methods, making it easier to interpret models built using PyTorch [22].

SHapley Additive exPlanations (SHAP) - SHAP is a package that offers a game-theoretic approach to explain the output of any machine learning model. As stated in the documentation, SHAP "connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions." Shapley values quantify each feature's contribution to the model's prediction for a given sample. SHAP explanations illustrate how strongly and in what direction each feature affects predictions, both locally and globally [21] [23].

Local Interpretable Model-agnostic Explanations (LIME) - LIME is designed to explain model predictions locally. It generates random points around the sample being explained, computes the model's output for these points, and trains a surrogate model on this output. Surrogate models are simple, interpretable models (e.g., linear regression) that approximate the predictions of the complex black-box model. Analyzing the surrogate model provides insights into the original model's behavior [23].

Transformer Debugger is a tool designed to facilitate the analysis and interpretation of transformer-based language models. It provides a detailed visualization of the attention mechanisms within these models, allowing researchers to inspect how different input tokens influence each other during the model's processing. This tool can highlight important patterns and dependencies in the data that may not be immediately obvious. By visualizing attention weights, the Transformer Debugger helps in understanding model decisions, identifying potential biases, and improving model performance. This added transparency is crucial for refining model architectures and training processes [24].

To understand better the tools and choose the effective one, we tried to do comparison analysis for the tools and here is our comparison table:

Name	**Explanation Type**	**Model types supported**	**Explanation potential**	**Speed**	**Visualization**
Captum	Global	Pytorch Models	Single feature	Slow	Yes
SHAP	Global and local	Any	Single feature or interactions	Fast (Tree Explainer Only)	Yes
LIME	Local	Any	Single feature or interactions	Medium	Yes
Transformer Debugger	Local and Global	Transformers	Attention patterns and weights	Fast (only for small models)	Yes

Table 1. Comparison of model explainability tools.

After analyzing all of the tools, we chose SHAP for several compelling

reasons. SHAP offers a unique combination of strengths that make it particularly well-suited for our research into the stability of language models under varying prompts.

3. METHODOLOGY

3.1. Definition and Importance of Stability in Language Models

Stability in the context of language models refers to the model's ability to generate consistent and reliable outputs when presented with variations of input prompts for the same task. Stability is crucial because it reflects the robustness of a model's understanding and processing capabilities. In practical applications, users expect language models to provide coherent and dependable responses even when the prompts are rephrased or slightly altered [20].

Stability can be quantitatively measured by evaluating the similarity of the outputs generated from different but semantically equivalent prompts. A highly stable model will produce outputs that are closely aligned regardless of minor variations in the input prompts. This concept is essential for ensuring the reliability and predictability of language models in real-world applications [19]. A stable language model inspires confidence in its ability to understand and interpret user inputs accurately, leading to a more seamless and satisfactory user experience [20].

Moreover, in safety-critical applications such as medical diagnosis or autonomous driving, where incorrect or inconsistent responses could have severe consequences, stability becomes paramount. A stable language model reduces the risk of erroneous outputs due to variations in input prompts, thereby enhancing the safety and effectiveness of such applications. And in our situation, stability in language models is of utmost importance when working with the Russian language, given its semantic variability, morphological complexity, cultural sensitivity, domain specificity, and user expectations. By ensuring stability, language models can effectively navigate these linguistic challenges and deliver dependable and meaningful interactions in Russian. Because of this, proposing a new evaluation metric for the benchmarks is crucial from different perspectives.

3.2. Experimental Setup

We presented the proposed methodology to test the hypothesis of the stability coefficient of language models to prompt changes and evaluated its potential implications for practical use and understanding of the issue. To do this, we conducted a series of experiments on tasks from the MERA benchmark.

We selected of tasks from the MERA benchmark typically 4 tasks (Chegeka, LCS, ruDetox, ruOpenBookQA). Sampling of task datasets to gather approximately 10 examples for initial hypothesis testing. After that generation of 10 variations of task texts for each selected task, ranging from brief to extensive.

To evaluate model performance, we used the Vikhr and TinyLlama, and etc. models with a temperature of 0 to eliminate random responses and focus on their generation results on different prompts. After that, we calculated the model performance in a traditional way which was shown in the official benchmark website. After that we calculated the model stability coefficient which will be shown in the 3.3 part of our work.

After calculating the coefficients, we applied interpretation methods, including SHAP, to visualize which parts of the input prompts the models paid attention to when forming responses. This approach allowed us to gain insights into how the model's attention and understanding could vary depending on prompt variations.

3.3. Calculation of the Model Stability Coefficient

The stability coefficient measures the consistency of a model's responses to different prompt variations. To calculate this coefficient, we followed these steps:

First, for each task and its variations we collected the model’s responces. After that, we used cosine similarity between prompts and generated answers for each of the prompt variations compare the model's responses to different prompts for the same task. The stability coefficient was calculated by averaging the similarity scores across all prompt variations for each task. A higher coefficient indicates more consistent responses, reflecting the model's stability in processing various input conditions.

And according to this our final formula to calculate the stability coefficient is like this:

$$S = \frac{1}{N}\sum_{i=1}^{N} Cosine\ Similarity\ (A_i, A_b)$$

Where:

- N is the number of prompt variations.
- A_i is the response generated by the model for the i- th prompt variation
- A_b is the response generated by the model for the baseline prompt.

After calculating the model stability coefficient, we used interpretation tools to visualize attention patterns and identify which parts of the prompts influenced the model's responses the most. This analysis provided additional context for understanding model stability and performance.

4. RESULTS AND ANALYSIS

In the results and analysis section, we present the findings of our experiments and delve into their implications. We begin by providing an overview of the model performance across different tasks and prompt variations. We then discuss the calculated stability coefficients and their significance in assessing the robustness of the language models. Through visualizations and interpretation of attention patterns using tools like SHAP, we gain insights into how the models process and respond to various input prompts. Furthermore, we examine any observed trends or patterns in model behavior and discuss potential avenues for future research and improvement. Overall, this section offers a comprehensive analysis of our experimental findings and their implications for the field of natural language processing.

Task name	Modality	Task Type	Output format	Class	Metric
CheGeka	Text	World Knowledge	Open Question	Examination	F1 / EM
LCS	Code	Algorithms	Multiclass Classification	Examination	Accuracy
ruDetox	Text	Ethics	Open questions	Diagnostic	Toxicity (STA) Content preservation (SIM) Fluency task (FL) J = J = STA * SIM * FL
ruOpenBook QA	Text	World Knowledge	Choosing an answer	Problematic	Avg. F1/ Accuracy

Table 2. Analyzed Tasks from the MERA benchmark [24]

4.1. Chegeka

CheGeKa is an open-domain question-answering dataset in Russian, consisting of QA pairs collected from the official Russian quiz database ChGK. It involves open-ended questions and is evaluated using the F1 score and Exact Match (EM) metrics. This task assesses a model's ability to provide accurate and complete answers to questions that require general knowledge about the world [24].

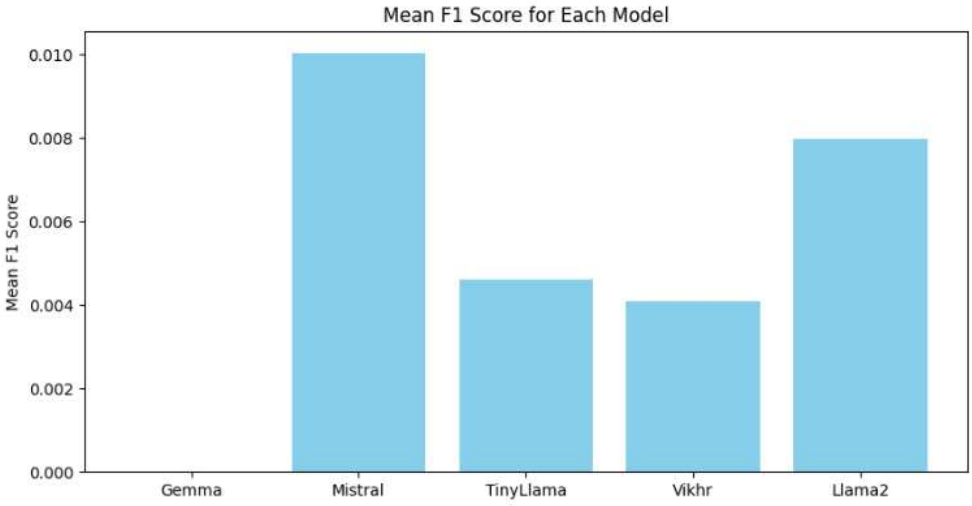

Figure 2. Chegeka Model answers result

Figure 3. Chegeka SHAP result for CheGeKa the best answer

In the SHAP result for CheGeKa presented in Pic 3, we observe the influence of different words on the model output, particularly focusing on the TinyLLama model's response.

The input prompt provided to the model is "Подумайте быстро! Что вы бы ответили на вопрос из категории 'любовь: продолжение этого фильма Евгения Матвеева называется? ответ:'," which translates to "Think quickly! What would you answer to the question from the category 'love: the continuation of this film by Evgeny Matveev is called? answer:"

The model's output response to this prompt is "Любовь в большом городе," which translates to "Love in the big city."

From the SHAP result, we can see which words in the input prompt had the most significant impact on the model's decision to generate this output. By visualizing the word attributions, we gain insights into the model's attention and understanding of the prompt. This allows us to interpret why the model selected "Любовь в большом городе" as the best answer given the input prompt.

For example, the model may have placed high importance on words like "любовь" (love) and "большом городе" (big city) in the prompt, indicating a strong association between these words and the correct response. Conversely, other words may have had minimal impact on the model's decision.

Figure 4. Chegeka SHAP result for CheGeKa the worst answer

In Figure 4, we observe the SHAP result for CheGeKa, focusing on the TinyLLama model's response, which represents the worst answer according to our evaluation criteria.

The input prompt provided to the model is: "Ваш вопрос из категории 'Любовь': Продолжение этого фильма Евгения Матвеева называется. Внимание у вас ограниченное время. Ответ:", which translates to "Your question from the category 'Love': The continuation of this film by Evgeny Matveev is called. Attention, you have limited time. Answer:"

Surprisingly, the model's output response to this prompt is: "1000 рублей. Вы можете заказать за любой фильмов, которые вы брали," which translates to "1000 rubles. You can order any films you've taken." This model's selection of "1000 рублей. Вы можете заказать за любой фильмов, которые вы брали" as the output response for the given prompt demonstrates a significant failure in comprehension and contextual understanding. This highlights the limitations and challenges associated with language model performance, particularly in accurately interpreting and responding to complex prompts.

4.2. LCS

LCS challenges language models to find the longest common subsequence between pairs of input strings. As a classic dynamic programming problem, LCS tests a model's ability to identify and apply efficient algorithmic approaches. The model's performance is evaluated based on accurately predicting the length of the longest subsequence shared by the two strings [24].

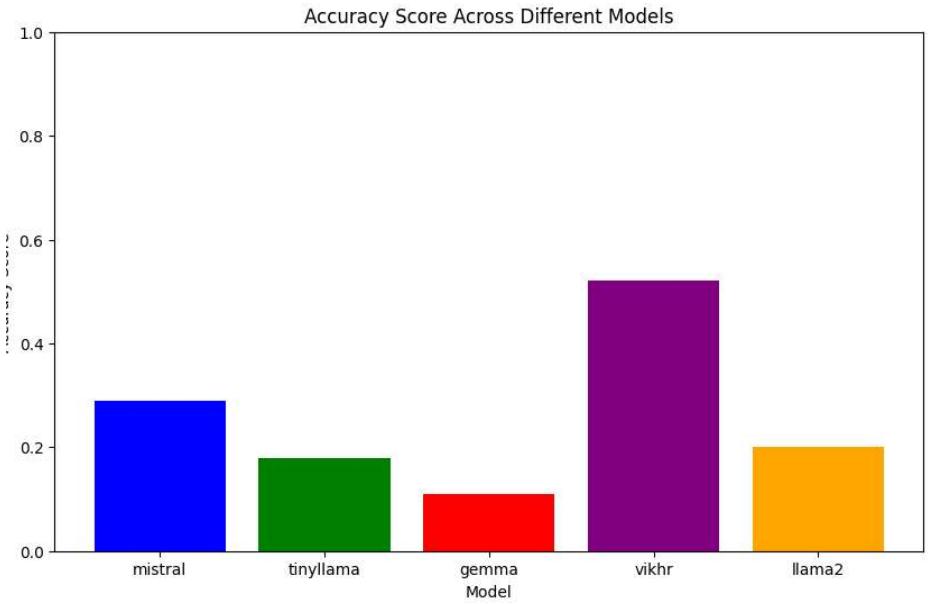

Figure 5. Model Performance Comparison on LCS Dataset

The chart reveals that the Vikhr model is the most effective at solving the LCS task, showcasing a high ability to find and apply efficient algorithmic solutions. Other models like Mistral and Llama2 also performed reasonably well, whereas Tinyllama and gemma had the lowest accuracy scores, indicating challenges in handling the LCS problem effectively.

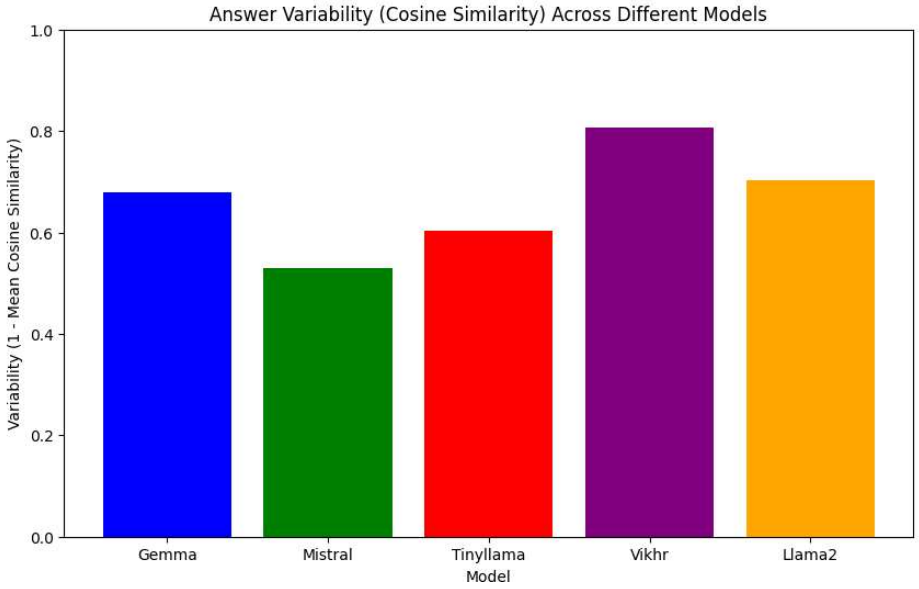

Figure 6. Answer Variability Comparison on LCS Dataset

The chart reveals that the Vikhr model produces the most variable answers, with the highest score in answer variability, followed by Llama2. This suggests that Vikhr and Llama2 may generate a wide range of responses, potentially capturing diverse aspects of the LCS problem but also indicating less consistency. Mistral, on

the other hand, shows the least variability, suggesting more consistent but potentially less diverse answers. Gemma and Tinyllama fall in between, showing moderate variability. This analysis helps in understanding the consistency and robustness of each model in generating answers for the LCS task.

Figure 7. SHAP result for the best answer of LCS task

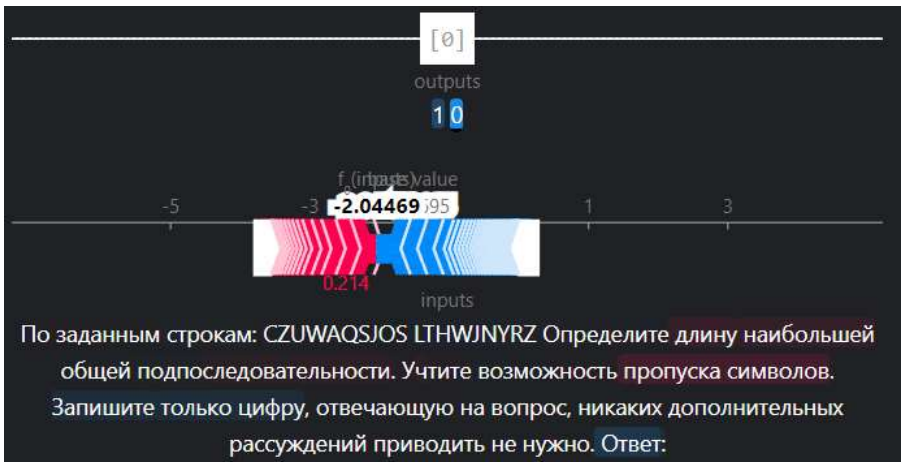

Figure 8. SHAP result for the worst answer of LCS task

In figure 7, and figure 8 we can see the SHAP results for the LCS task. Red Bars: Features (characters in the sequences) that contribute positively to the model's output (moving the output towards the actual prediction), and blue bars are features that contribute negatively (moving the output away from the actual prediction).

The baseline value which represents the average model output over the training set (around -1.77501 here, though this interpretation depends on how SHAP values are calculated in the specific context).

In figure 8, the task involves determining the length of the LCS for the strings

"CZUWAQSJOS" and "LTHWJNYRZ". The model predicted an LCS length of 10, which is significantly incorrect. The comparison highlights that while the model's prediction mechanisms are influenced by specific character positions, the magnitude and distribution of SHAP values vary significantly between predictions. Inaccurate predictions, as seen in Figure 8, often result from overemphasis on incorrect alignments within the sequences. This underscores the importance of refining the model to balance contributions and improve accuracy in sequence alignment tasks. Further model tuning and advanced feature engineering could mitigate such errors and enhance prediction reliability.

4.3. ruDetox

ruDetox is a diagnostic dataset for evaluating models' ability to detoxify offensive Russian language while preserving meaning and fluency. The task involves transforming input sentences containing toxic or abusive content into more neutral and polite rephrasings. Models are assessed on their detoxification effectiveness, semantic preservation, and output fluency [24].

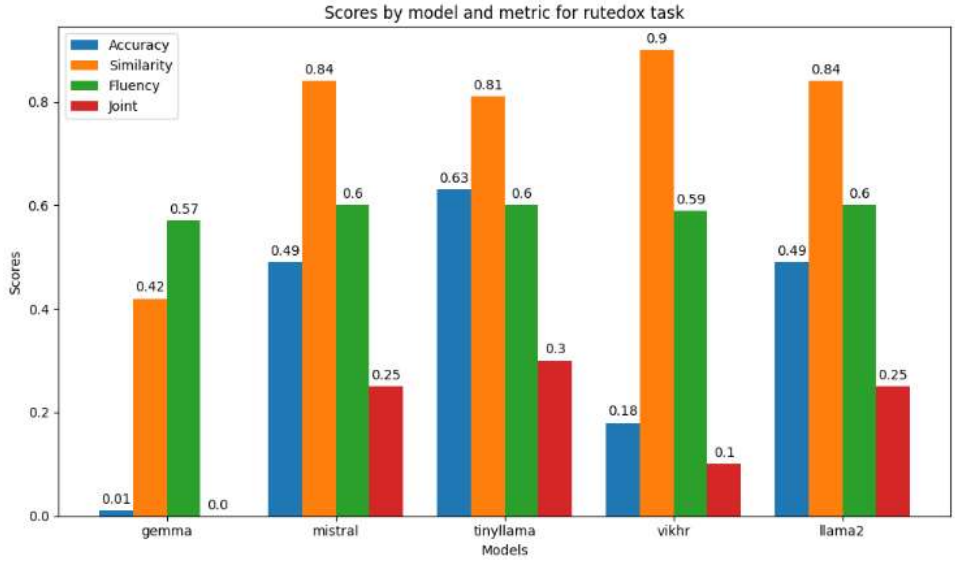

Figure 9. Model Performance Comparison on RuDetox Dataset

This bar plot shows the performance of five different models—Gemma, Mistral, TinyLlama, Vikhr, and Llama2—on the ruDetox dataset across four evaluation metrics: Style Accuracy, Similarity, Fluency, and Joint Score. Each metric provides insight into different aspects of the model's performance in detoxifying text while maintaining the original meaning and fluency.

From the plot, we observe:

Gemma has low performance across all metrics, particularly in style accuracy and joint score. Mistral and Llama2 show similar performance, with moderate scores in style accuracy, high similarity, and fluency. TinyLlama achieves the highest style accuracy and joint score, indicating a well-rounded performance. Vikhr excels in similarity but falls behind in style accuracy and joint score.

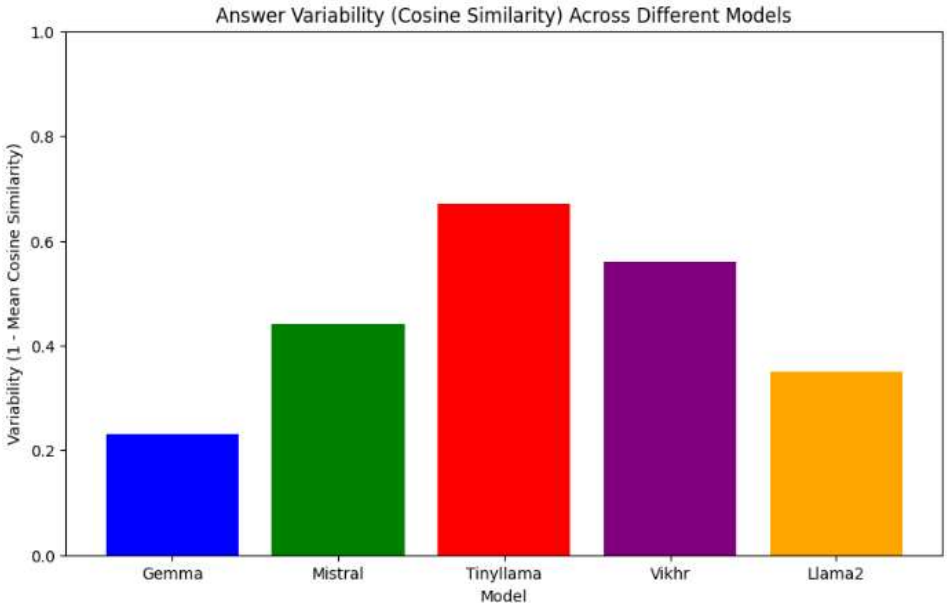

Figure 10. Answer Variability Across Different Models (rutedox)

This graph illustrates the variability in answers generated by different models, measured using cosine similarity. Cosine similarity values range from 0 to 1, where higher values indicate higher similarity between generated answers and prompts. Models Vikhr and Tinyllama demonstrate relatively higher variability, while Mistral and Llama2 exhibit similar variability.

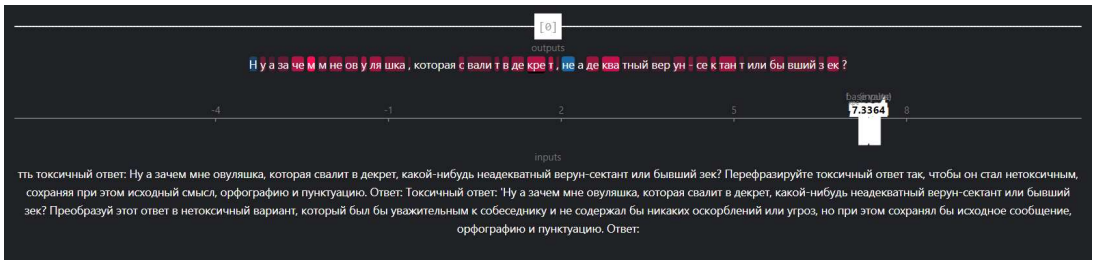

Figure 11. SHAP result for the worst answer of rutedox task

In figure 11, the SHAP results for the worst answer of rutedox task is shown. The analysis indicates a baseline value of approximately 7.3364. This value represents the average prediction of the model over the training set. The figure shows

contributions from specific words and phrases that influence the model's decision to label the comment as toxic. Positive contributions (highlighted in red) significantly push the output towards toxicity, while negative contributions (highlighted in blue) have a lesser effect in this context.

Figure 12. SHAP result for the best answer of rutedox task

In Figure 12, the analysis is related to a natural language processing (NLP) task, specifically detecting toxicity in text. The input text in Russian: "Ну а зачем мне овуляшка, которая свалит в декрет, какой-нибудь недавний верун-сектант или бывший зек?" Translated, it roughly means: "Why would I need a woman who would go on maternity leave, some recent sectarian convert, or a former convict?" The output seems to be a masked or transformed version of the input text, possibly as a part of generating a non-toxic response: "Ну а зачем мне овуляшка, которая свалит в декрет, и не адекватный веру ни сектант и или бывший зек?"

The output includes highlighted parts indicating the model's focus. The SHAP values are visualized on a scale where the impact of each part of the text on the model's prediction is shown. The color intensity and position (positive or negative side) indicate how each part of the text contributes to the predicted toxicity score.

4.4. ruOpenBookQA

ruOpenBookQA is a Russian multiple-choice question-answering dataset consisting of elementary-level science questions. The questions test understanding and application of core scientific facts, as well as related common-sense reasoning. To simulate an open-book exam, models are provided with relevant textual resources and must select the correct answer from several choices.

Inspired by the English OpenBookQA dataset, ruOpenBookQA aims to assess language models' factual knowledge, logical reasoning, and ability to apply information to novel scenarios. By grounding the questions in a "book" of core scientific facts, this task constrains the relevant knowledge and places emphasis on inferential reasoning skills. Since the dataset is designed to be challenging for retrieval-based and co-occurrence algorithms, strong performance requires going beyond surface-level pattern matching to achieve genuine understanding and reasoning. Models that excel at ruOpenBookQA can be seen as having more robust and generalizable language comprehension abilities.

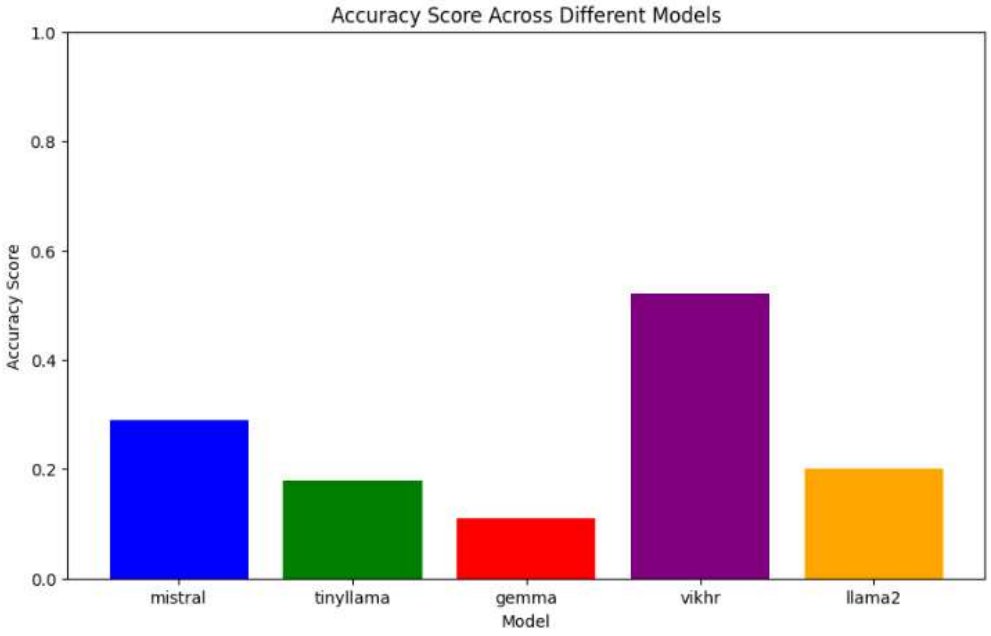

Figure 13. Accuracy Scores of Different Models (ruOpenBookQA)

This bar graph illustrates the accuracy scores achieved by different models: Mistral, Tinyllama, Gemma, Vikhr, and Llama2. The accuracy score is a measure of how accurately each model's generated answers match the expected answers. From the graph, it is evident that Vikhr achieved the highest accuracy score among the models, followed by Mistral, Llama2, Tinyllama, and Gemma. The variations in accuracy scores among the models highlight differences in their performance in generating answers.

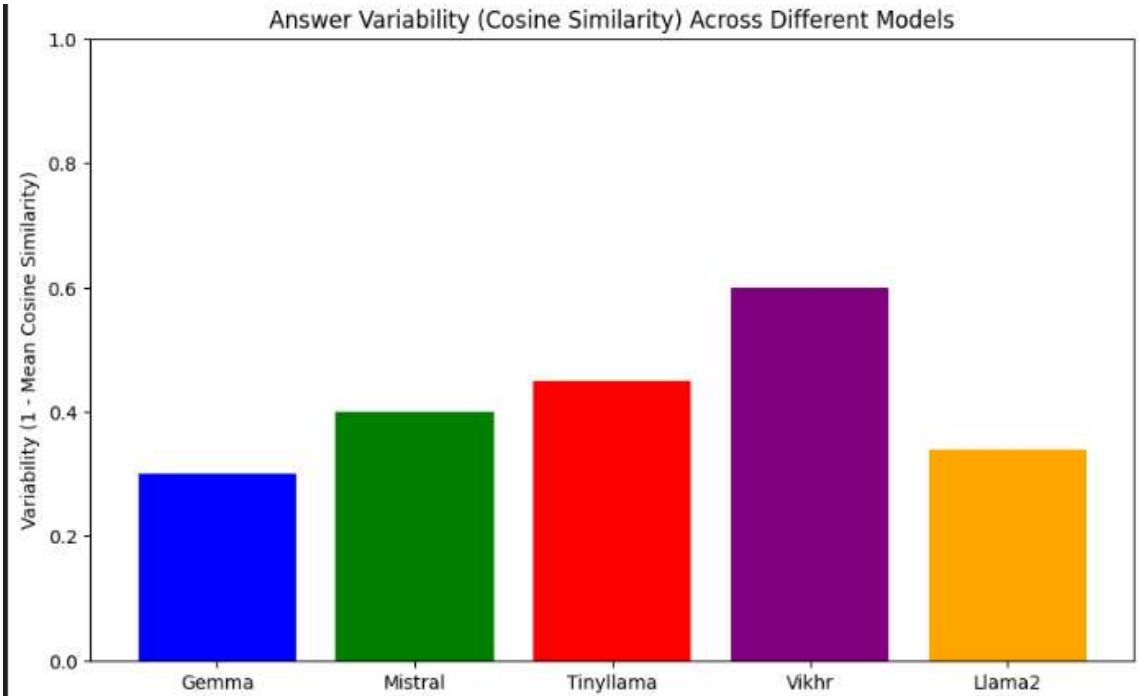

Figure 14. Answer Variability (Cosine Similarity) Across Different Models (ruOpenBookQA)

This plot illustrates the answer variability for five different language models (Gemma, Mistral, Tinyllama, Vikhr, and Llama2) by calculating the cosine similarity between their generated responses and the corresponding prompts. The variability score, defined as 1 − mean cosine similarity indicates how much the model's generated answers deviate from the prompts.

Figure 15. SHAP result for the best answer of ruOpenBookQA task

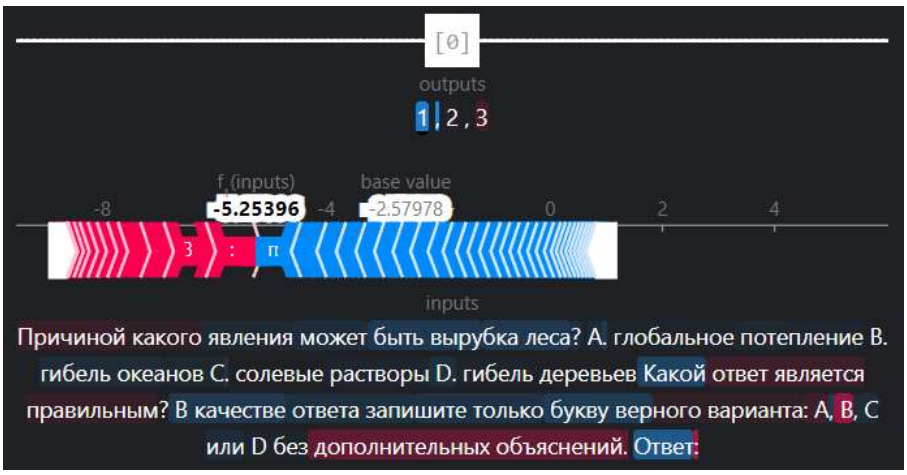

Figure 16. SHAP result for the worst answer of ruOpenBookQA task

Figures 15 and 16 are visualizing SHAP (SHapley Additive exPlanations) analysis results for different answers to the same question of the ruOpenBookQA. In Figure 15, the longest bar with a negative value (-3.64818) suggests that a particular feature or set of features had a significant negative contribution to the output value of 2. However, the overall output value is still positive, implying that other features counterbalanced this negative contribution.

In contrast, in Figure 16, the longest bar (-5.25396) has a more substantial negative contribution compared to Figure 15. Additionally, the base value (-2.57978) is also negative, suggesting that the overall input features had a more negative impact on the output value of 1, 2, 3, which the model considers a less favorable answer.

5. CONCLUSION

In conclusion, our work has contributed to enhancing the evaluation and understanding of language models by introducing the concept of a stability coefficient. Through our experiments and analysis, we have demonstrated the importance of considering model stability in response to prompt variations, which traditional evaluation metrics may overlook. By proposing a stability coefficient, we provide a quantitative measure to assess the consistency and reliability of language models across different input prompts.

Our findings highlight the significance of stable model behavior in real-world applications, where users expect consistent and dependable responses regardless of how questions are phrased. The stability coefficient offers a valuable tool for researchers and developers to gauge the robustness of language models and make informed decisions about their suitability for specific tasks and scenarios.

Moving forward, further research and development efforts should focus on refining and standardizing methods for calculating and interpreting the stability coefficient. Additionally, exploring ways to improve model stability through fine-tuning and architecture enhancements will be essential for advancing the field of natural language processing.

References:

1. Fenogenova, A., Tikhonova, M., Mikhailov, V., et al. (2022). Russian SuperGLUE 1.1: Revising the Lessons Not Learned by Russian NLP models. DOI: 10.28995/2075-7182-2021-20-XX-XX

2. Wang, A., Wang, X., Ji, X., et al. (2023). Assessing and optimizing large language models on spondyloarthritis multi-choice question answering (SpAMCQA): study protocol for a bilingual evaluation benchmark. DOI: 10.21203/rs.3.rs-3625354/v1

3. Panchenko, A., et al. (2018). RUSSE'2018: a shared task on word sense induction for the Russian language. arXiv preprint arXiv:1803.05795

4. Ruder, S. (2021). Challenges and Opportunities in NLP Benchmarking. URL: https://www.ruder.io/nlp-benchmarking/

5. Elov, B. B., Khamroeva, Sh. M., Xusainova, Z. Y. (2023). The pipeline processing of NLP. E3S Web of Conferences 413, 03011. DOI: https://doi.org/10.1051/e3sconf/202341303011

6. Song, L., Zhang, J., Cheng, L., et al. (2023). NLPBench: Evaluating Large Language Models on Solving NLP Problems.

7. Storks, S., Gao, Q., Chai, J. Y. (2019). Recent Advances in Natural Language Inference: A Survey of Benchmarks. DOI: https://doi.org/10.48550/arXiv.1904.01172

8. Iazykova, T., Kapelyushnik, D., Bystrova, O., Kutuzov, A. (2021). Unreasonable Effectiveness of Rule-Based Heuristics in Solving Russian SuperGLUE Tasks. arXiv: 2105.01192v1 [cs.CL]

9. Shavrina, T., Shapovalova, O. (2017). To the methodology of corpus construction for machine learning: "Taiga" syntax tree corpus and parser. Proceedings of "CORPORA-2017" International Conference. 2017.

10. Dagan, I., Glickman, O., Magnini, B. (2005). The pascal recognising textual entailment challenge. Machine Learning Challenges Workshop. Springer, Berlin, Heidelberg.11. Panchenko, Alexander, et al. "RUSSE'2018: a shared task on word sense induction for the Russian language." arXiv preprint arXiv:1803.05795

(2018).

11. Liang, P., et al. (2023). Holistic Evaluation of Language Models. arXiv:2211.09110

12. Stanford University – Human Centered Artificial Intelligence. (2022). Language Models Are Changing AI. We Need to Understand Them. URL: https://hai.stanford.edu/news/language-models-are-changing-ai-we-need-understand-them

13. Shavrina, T., et al. (2020). RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark. arXiv:2010.15925

14. Srivastava, A., et al. (2022). Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. arXiv:2206.04615

15. Colombo, P., et al. (2022). What are the best systems? New Perspectives on NLP Benchmarking. Advances in Neural Information Processing Systems 35 (NeurIPS 2022) Main Conference Track

16. Wołk, K. (2015). Neural-Based machine translation for the medical text domain. Based on European Medicines Agency leaflet texts. Procedia Computer Science 64:2-9.

17. Lavie, A., & Agarwal, A. (2007). METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In Proceedings of the Second Workshop on Statistical Machine Translation (pp. 228–231).

18. Prather J. et al. Interactions with Prompt Problems: A New Way to Teach Programming with Large Language Models // arXiv:2401.10759 [cs]. - 2024.

19. Wolf, T., et al. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771

20. Gao, T., Fisch, A., & Chen, D. (2021). Making Pre-trained Language Models Better Few-shot Learners. Association for Computational Linguistics (ACL)

21. Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. Association for Computational Linguistics (ACL).

22. Introduction to Captum — A Model Interpretability Library for PyTorch [Electronic resource]. URL: https://medium.com/pytorch/introduction-to-captum-a-model-interpretability-library-for-pytorch-d236592d8afa

23. Model Explainability: How to Choose the Right Tool [Electronic resource]. URL: https://medium.com/ing-blog/model-explainability-how-to-choose-the-right-tool-6c5eabd1a46a.

24. Tasks — MERA [Electronic resource]. URL: https://mera.a-ai.ru/en/tasks.

Appendices:

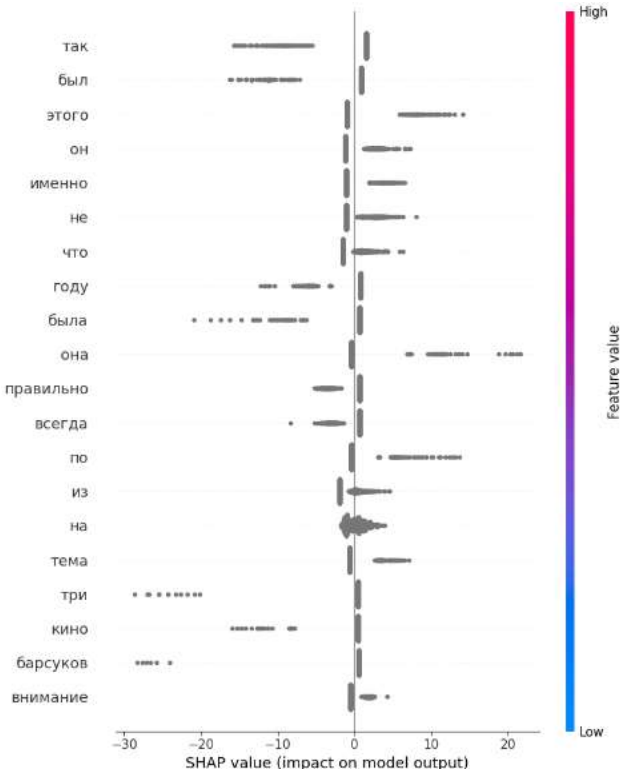

Fig 1. The SHAP result of the important words for Vikhr model answers

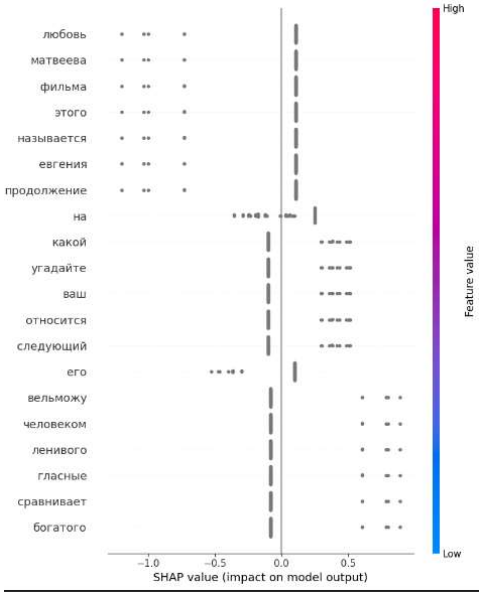

Fig 2. The SHAP result of the important words for TinyLLama model answers

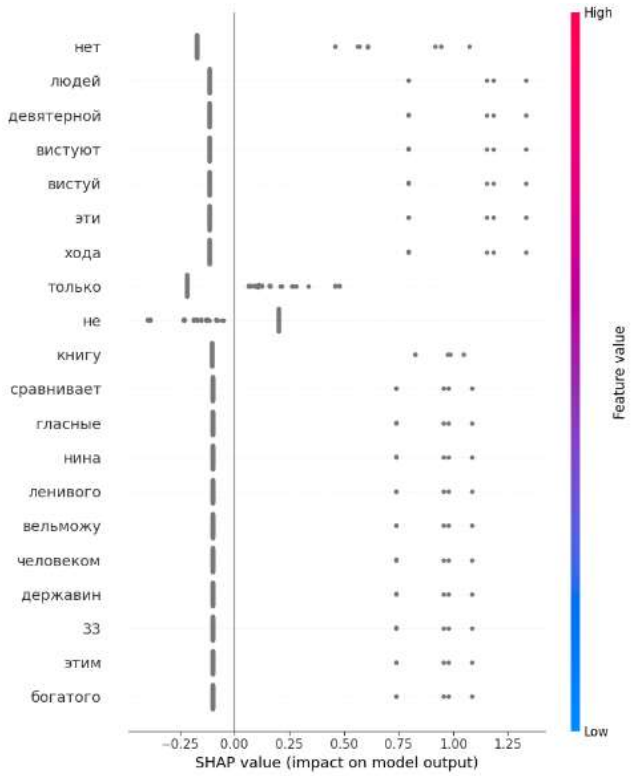

Fig 3. The SHAP result of the important words for Gemma model answers

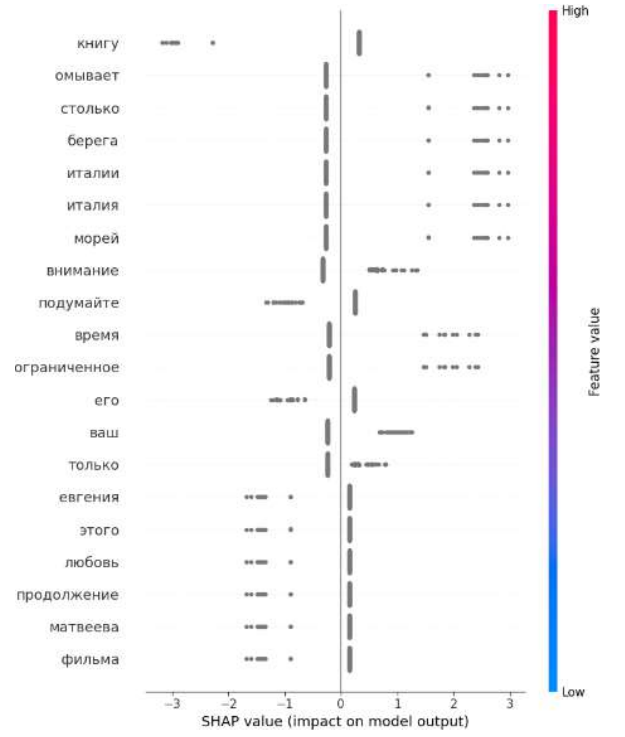

Fig 4. The SHAP result of the important words for Mistral model answers

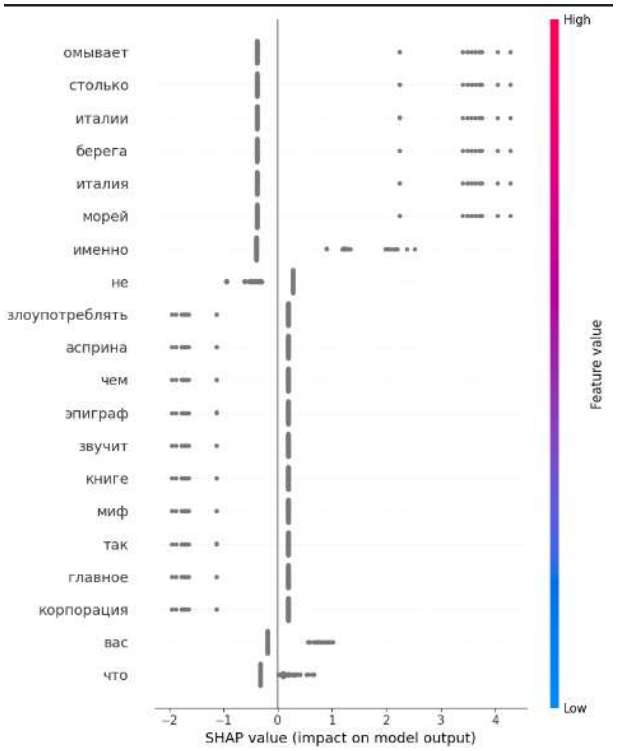

Fig 5. The SHAP result of the important words for Llama 2 model answers