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. 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. 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].

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 output 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 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 parts of our work.

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

2.3. 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:

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

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.
- P is answering probability in the prompt.

The result is then multiplied by the probability of the answer, which represents the likelihood of the response being correct based on the token probabilities.

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.

3. 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]

3.1. 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].

We first calculated the f1 score of the model outputs which was shown in the official MERA benchmark leaderboard [24]. Here you can see the results:

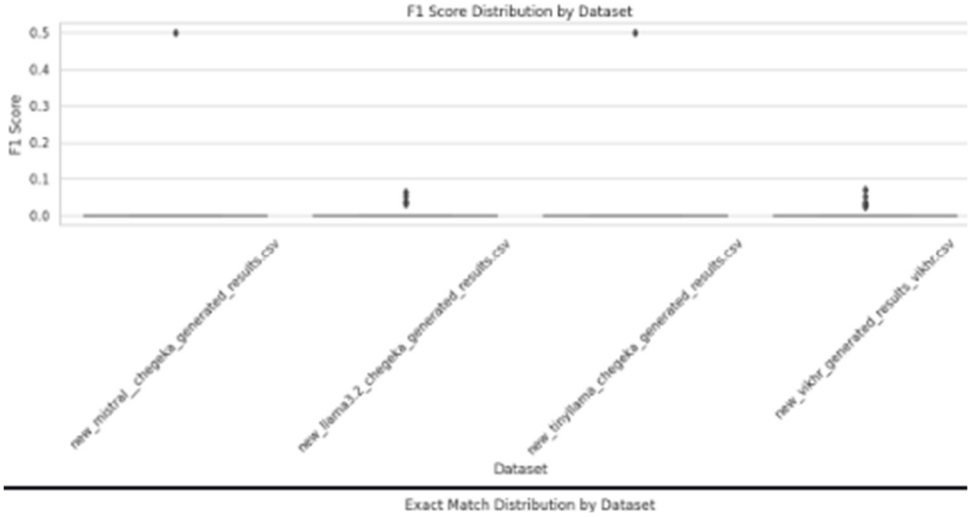

Figure 2. Chegeka f1 score answers result

Before calculating the stability coefficient, we tried to understand and evaluate the combined score of the quality of generated responses to questions using a combination of semantic similarity, relevance to the question, and linguistic accuracy.

To achieve this, we first encode the question, reference answer, and generated answer into their respective embeddings using a pre-trained model. This step enables us to represent the textual data in a numerical format that captures their semantic meanings. We then compute the semantic similarity between the reference answer and the generated answer by applying cosine similarity. This metric helps us understand how closely the generated response aligns with the expected answer in terms of content and meaning.

In addition to these semantic evaluations, we also utilize the BLEU score to gauge the linguistic accuracy of the generated response. BLEU is a metric commonly

used in machine translation that compares the generated text to a reference text based on the overlap of n-grams. This score provides insights into the fluency and correctness of the generated answer.

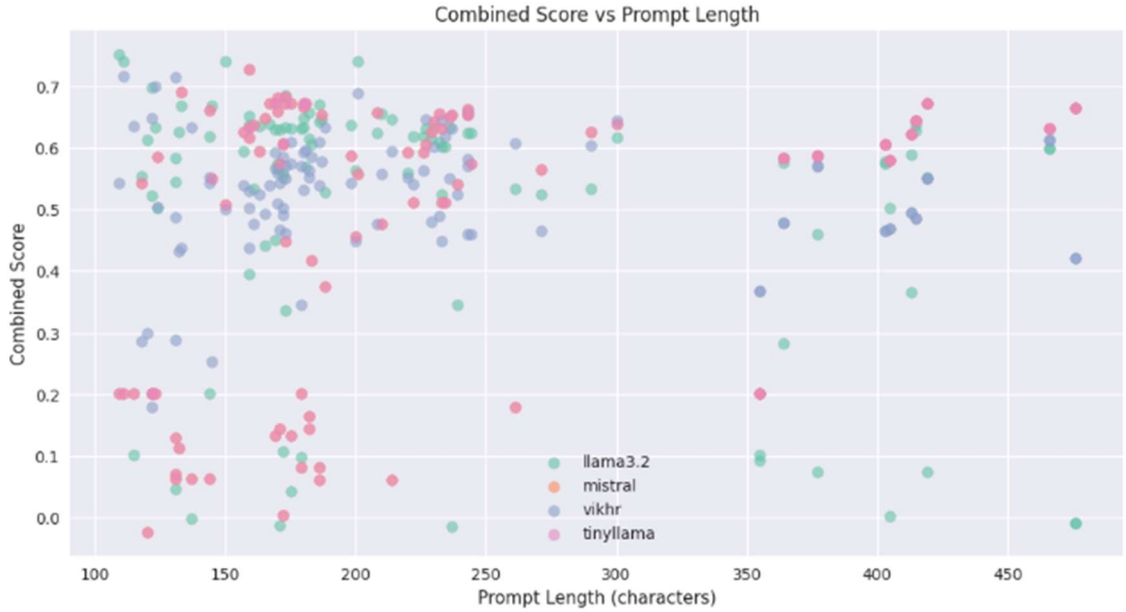

Figure 3. Impact of the prompt length to Combined score

In figure 3 we can see the relationship between the combined score (a composite metric reflecting both the quality and length of the generated text) and the prompt length (the number of characters in the input prompt). Each data point represents a individual sample or example. The plot reveals that there is a generally positive correlation between prompt length and combined score, but with considerable variability. This suggests that while longer prompts tend to produce higher quality outputs, there are other factors beyond just prompt length that influence the model's performance.

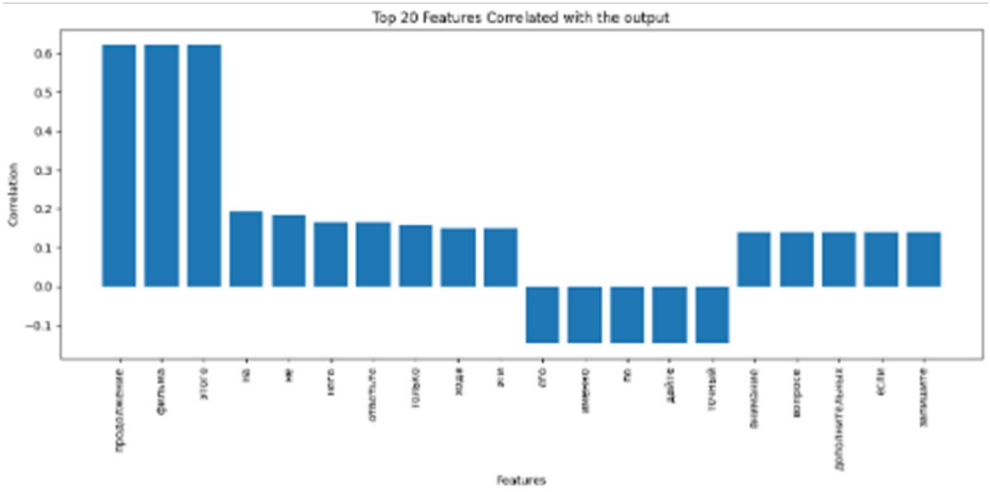

Figure 4. Top 20 features of the outputs

This bar chart depicts the top 20 features (or input variables) that are most strongly correlated with the model's output. The correlation coefficients are shown on the y-axis, with the most positively correlated features at the top and the most negatively correlated features at the bottom. This information can provide insights into which input variables are the most important predictors of the model's output, which can be valuable for feature selection, model interpretability, and further development of the system.

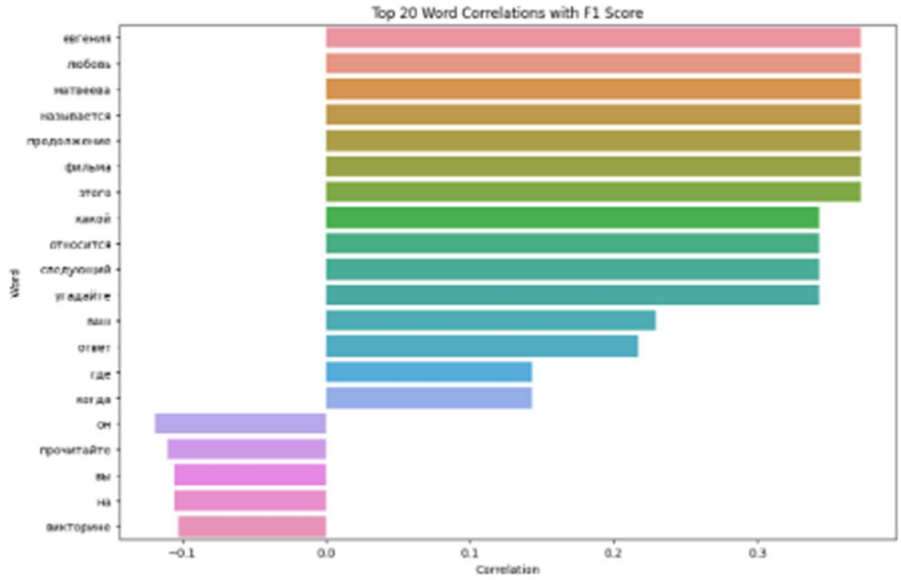

Figure 5. Top 20-word correlation with f1-score

This graph shows the top 20 word features that are most correlated with the F1 score, which is a common metric used to evaluate the performance of a machine learning model. The correlation coefficients are displayed on the y-axis, with the most positively correlated words at the top and the most negatively correlated words

at the bottom. This information can be useful for understanding which specific words or linguistic features are the most predictive of the model's performance.

And finally, here you can see the overall results of models on the our evaluation metrics: Stability Coefficient.

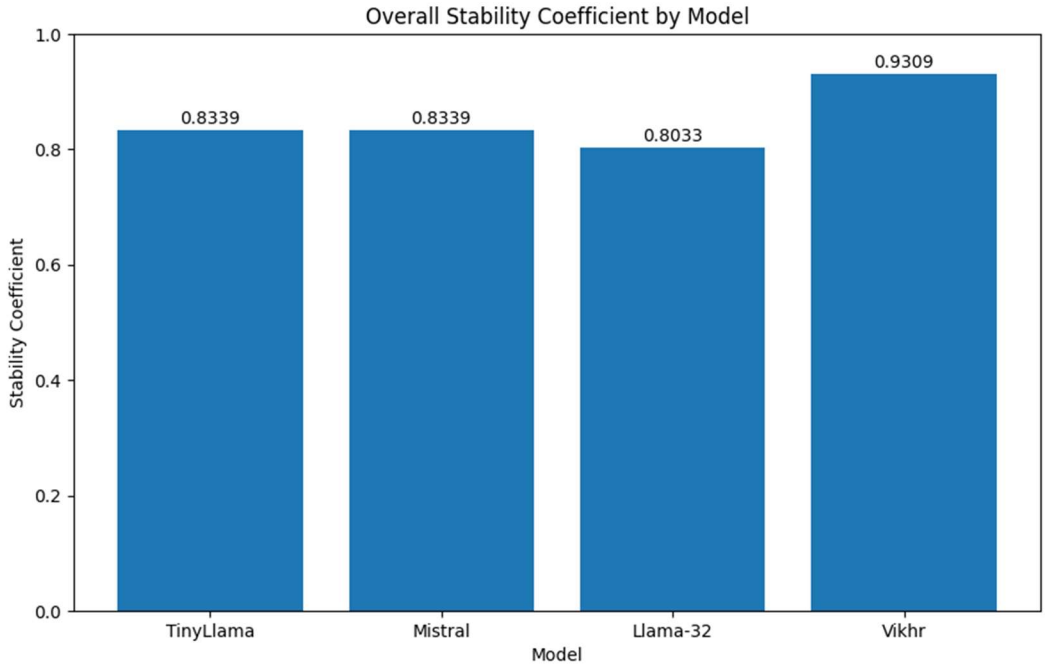

Figure 6. Overall Stability Coefficient

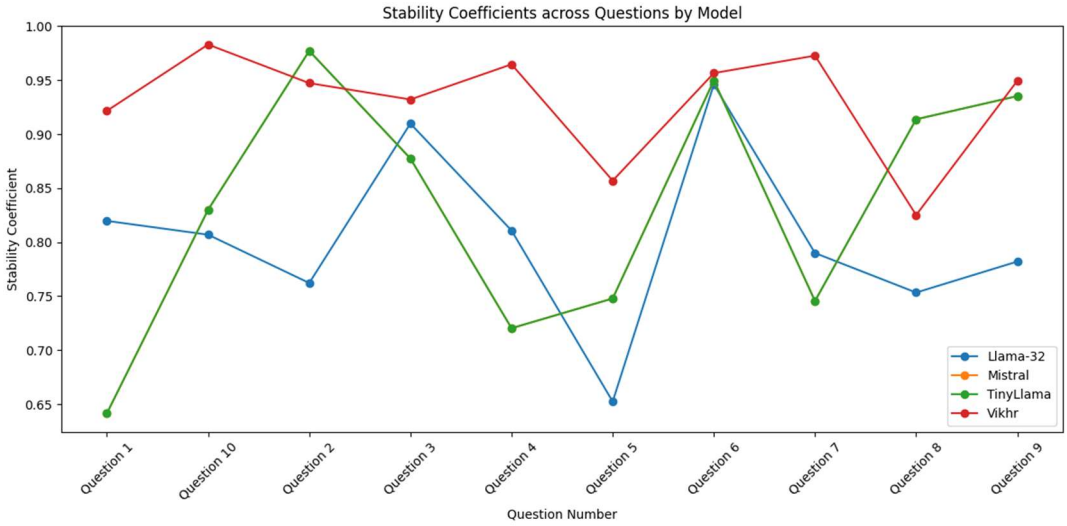

Figure 7. Overall Stability Coefficient for the different questions on Chegeka

From the given 2 graphs we can give a conclusion that Vikhr appears to be the most stable model overall, with both the highest average stability and the most consistent performance across questions. TinyLlama and Mistral perform similarly, with moderate stability and high variability. Llama-32 shows the lowest overall stability but with less variability than TinyLlama and Mistral.

In addition to the calculation of model answers, we looked at the SHAP and LIME model explanation results to see how models are affected by the prompts.

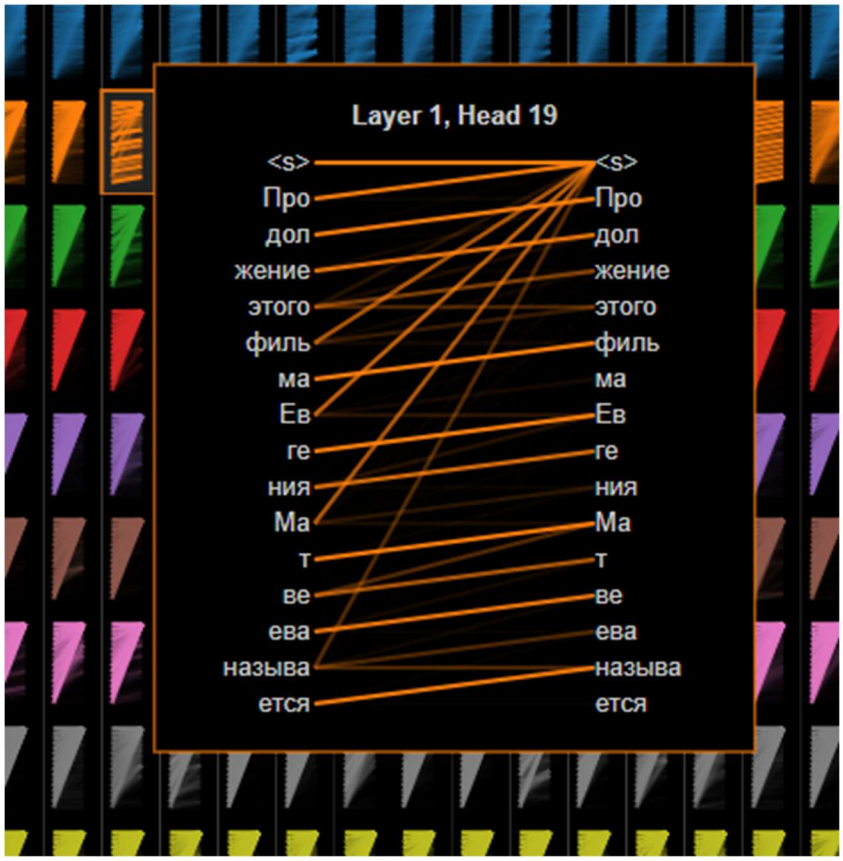

Figure 8. Chegeka SHAP result for CheGeKa the best answer

Here we can see the graph which is produced from the Bertviz to see how the model has attention for the different words and the parts of the sentence.

Figure 8. Chegeka SHAP result for CheGeKa the best answer

In the SHAP result for CheGeKa presented in Figure 8, we observe the influence of different words on the model output, particularly focusing on the TinyLLama model's response.

The prompt input 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 the question from the category 'love: the continuation of this film by Evgeny Matveev is called? answer:"

The model’s 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 9. Chegeka SHAP result for CheGeKa the worst answer

In Figure 9, 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.

3.2. 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].

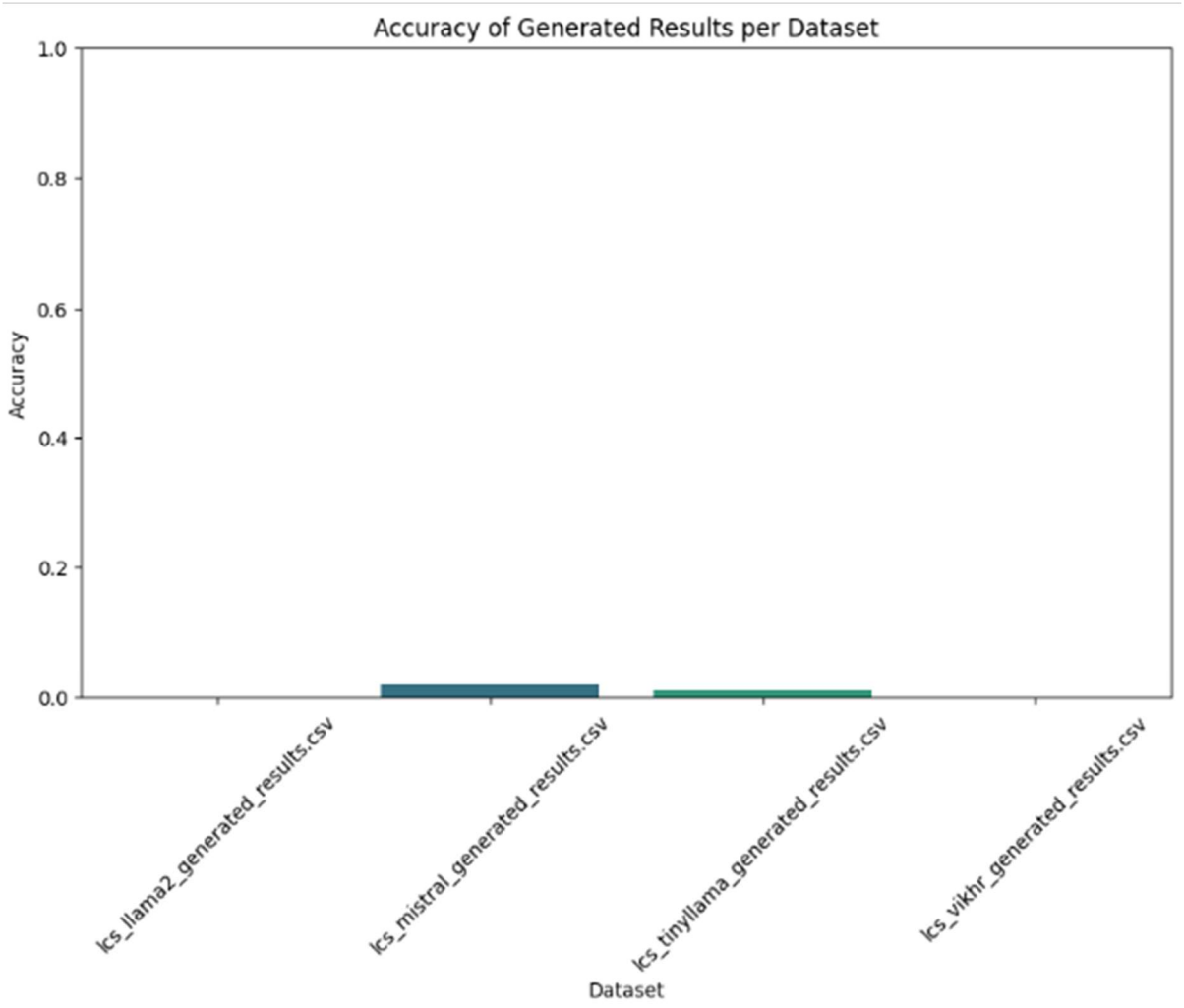

Figure 10. Model Performance Comparison on LCS Dataset

The chart reveals that the Mistral model is the most effective at solving the LCS task, showcasing a high ability to find and apply efficient algorithmic solutions. Other models like Vikhr and Llama3.2 also performed reasonably well, whereas Tinyllama had the lowest accuracy scores, indicating challenges in handling the LCS problem effectively.

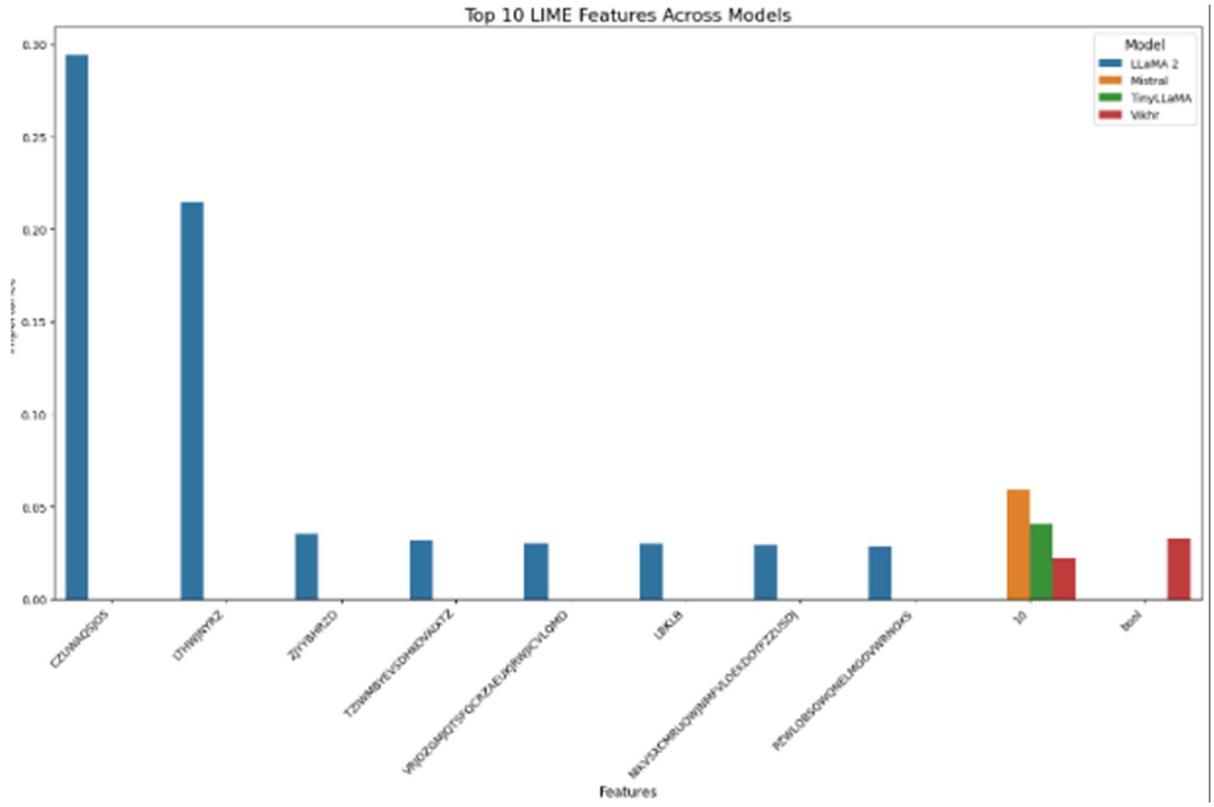

Figure 11. LIME features

These four smaller graphs display the top influential features for four different language models. The features are ranked by their correlation coefficients, shown on the x-axis, with the most positively correlated features on the right and the most negatively correlated features on the left. This provides insights into the specific linguistic and semantic cues that are most important for each model's performance, which can inform model development, feature selection, and fine-tuning.

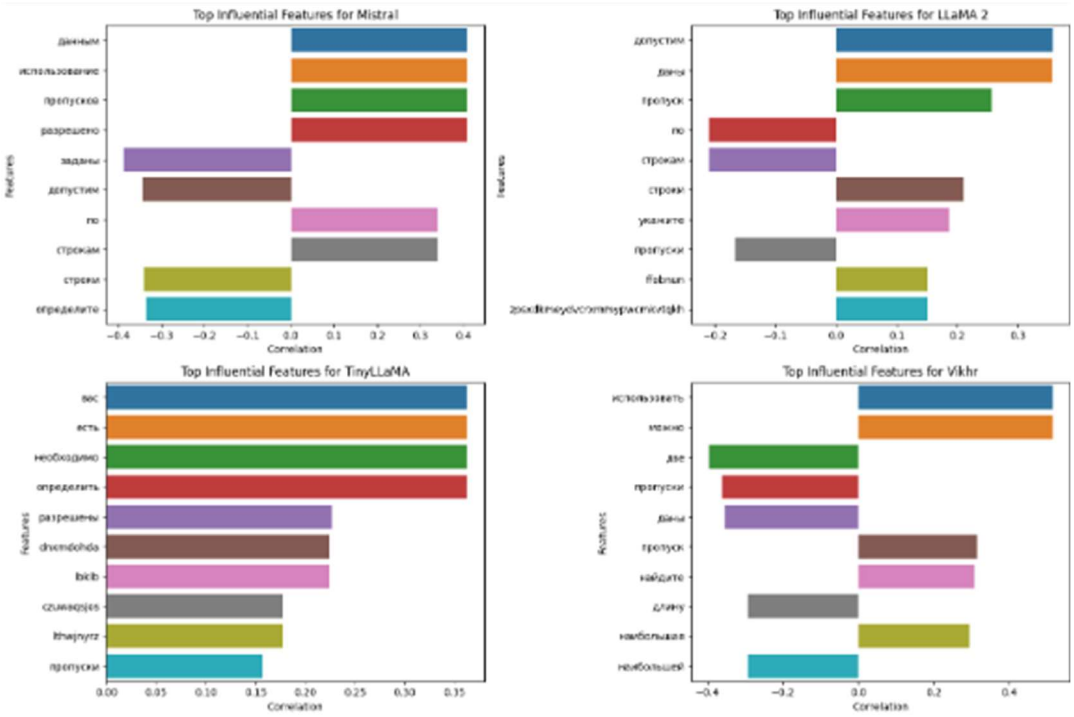

Figure 12. Top influential features for the models

This graph shows the top 10 most important features, as determined by the LIME (Local Interpretable Model-Agnostic Explanations) method, for various language models. The y-axis lists the features, while the x-axis shows the correlation coefficient of each feature with the model's output. The colored bars represent the different models, allowing for a comparison of the top features across the models. This information can be useful for understanding which linguistic or semantic elements are the most predictive for the models' performances.

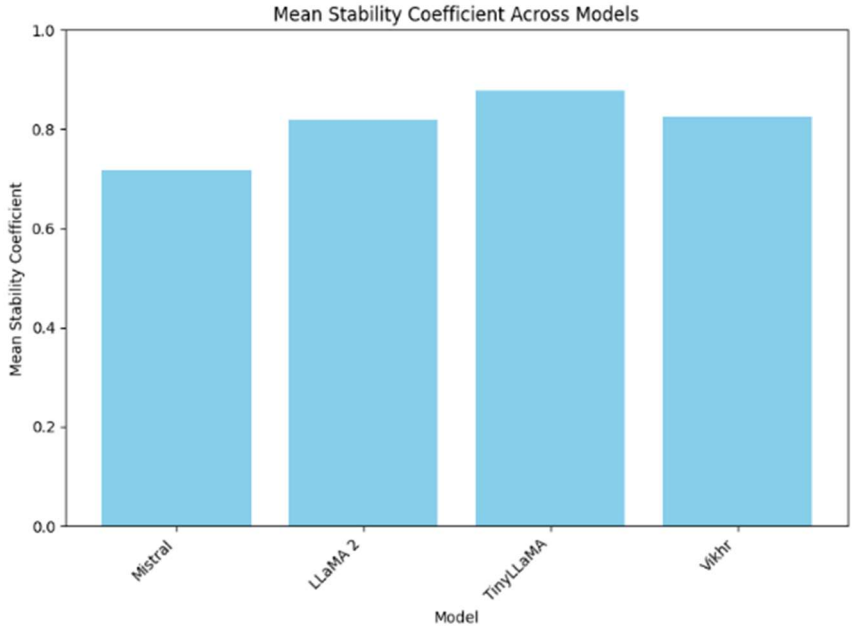

Figure 13. Stability Coefficient Across models

The chart reveals that the Llama3.2 model produces the most variable answers, with the highest score in answer variability, followed by Vikhr. This suggests that Vikhr 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. 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 14. SHAP result for the best answer of LCS task

Figure 15. 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.

3.3. 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 rephrasing's. Models are assessed on their detoxification effectiveness, semantic preservation, and output fluency [24].

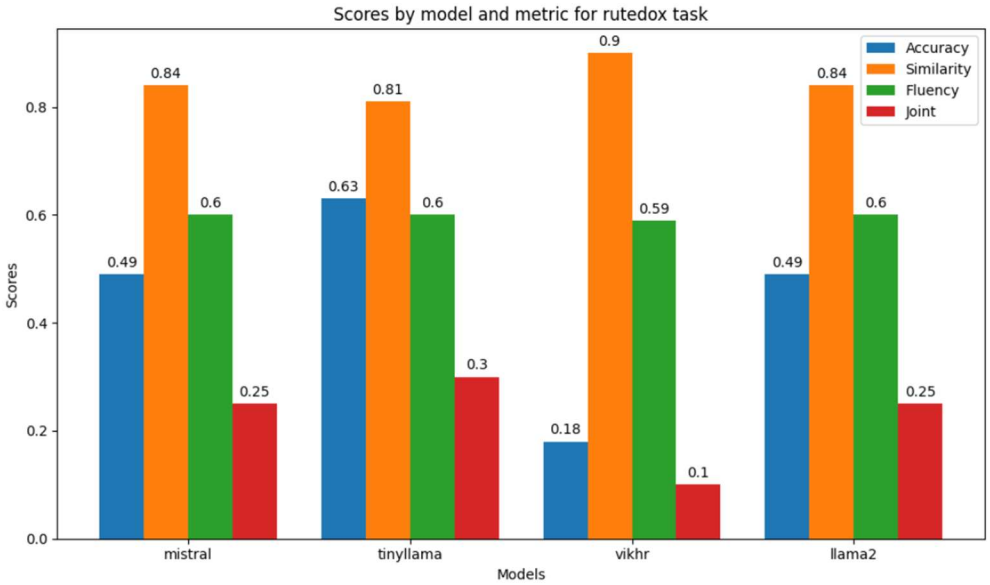

Figure 16. Model Performance Comparison on RuDetox Dataset

This bar plot shows the performance of five different models—Llama3.2, Mistral, TinyLlama, Vikhr — 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: Llama3.2 has low performance across all metrics, particularly in style accuracy and joint score. Mistral shows 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.

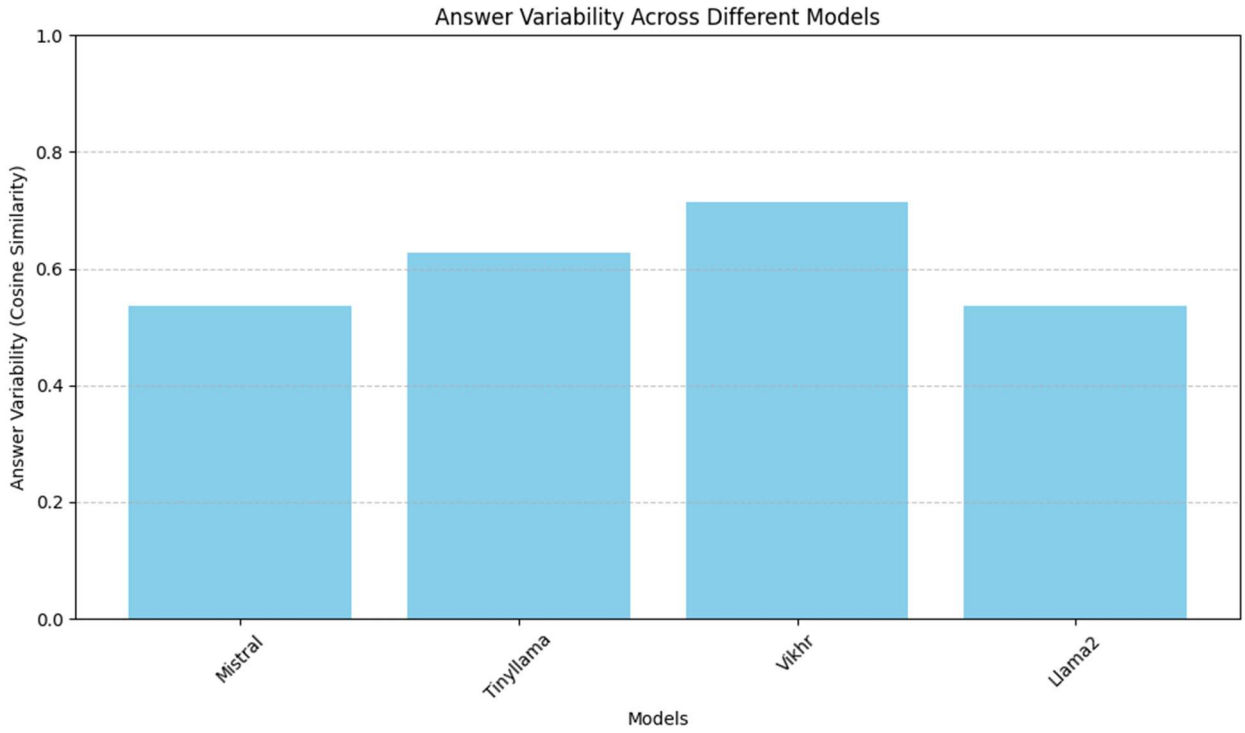

Figure 17. Answer Variability Across Different Models (rutedox)

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 Llama3.2 exhibit similar variability.

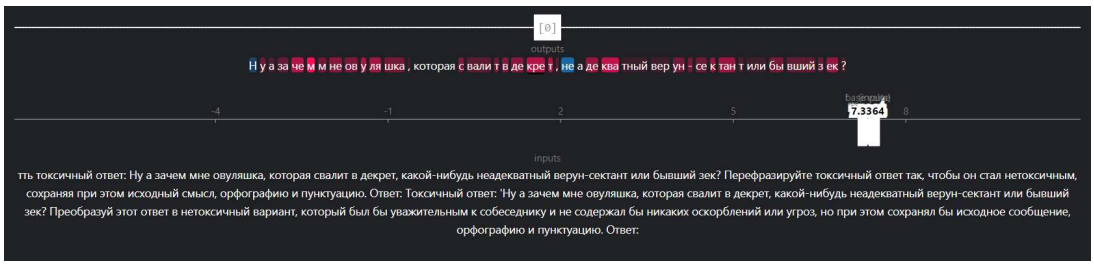

Figure 18. SHAP result for the worst answer of rutedox task

In figure 18, 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 19. SHAP result for the best answer of rutedox task

In Figure 19, 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.

3.4. 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.

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, Llama3.2, Tinyllama. The variations in accuracy scores among the models highlight differences in their performance in generating answers.

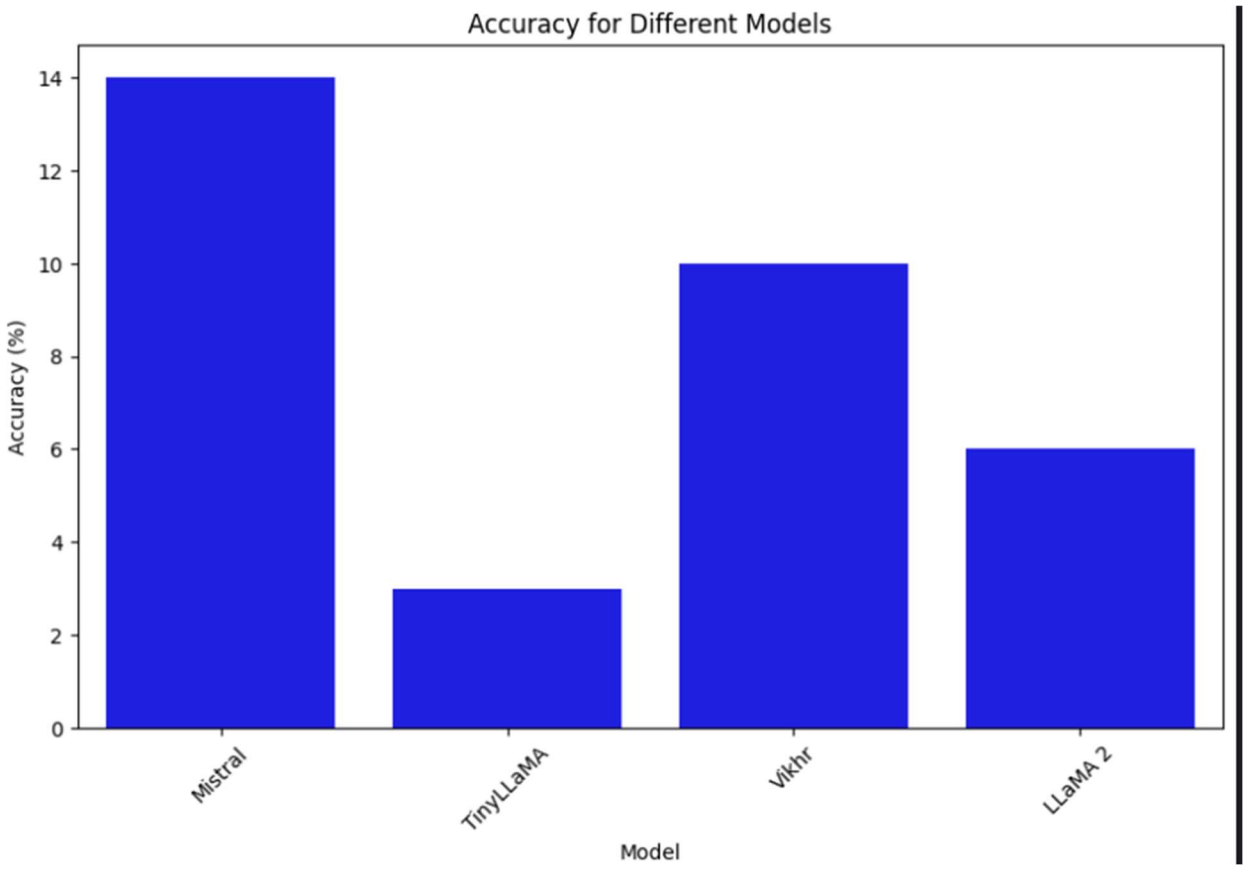

Figure 20. Accuracy calculation Across Different Models (ruOpenBookQA)

This plot illustrates the answer variability for five different language models (Llama3.2, Mistral, Tinyllama, Vikhr) 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.

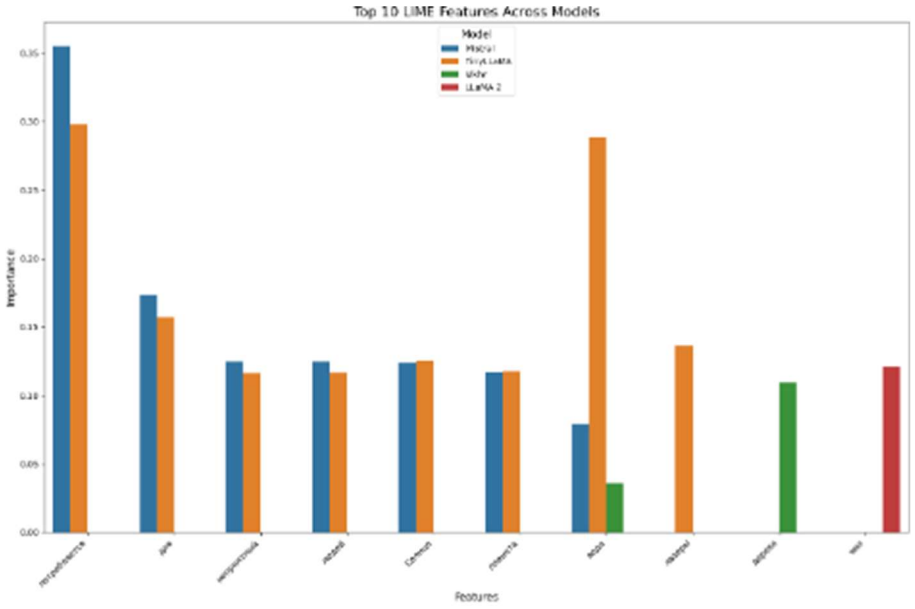

Figure 21. Top 10 LIME features across models (ruOpenBookQA)

Figure 22. SHAP result for the best answer of ruOpenBookQA task

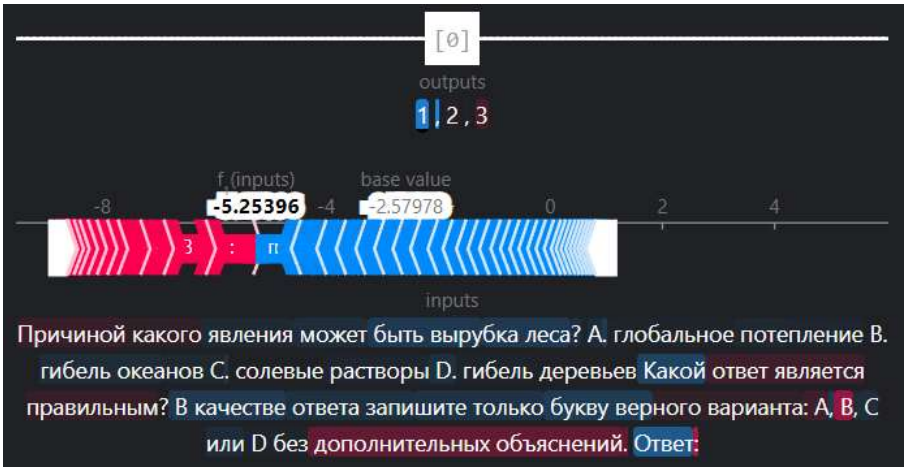

Figure 23. SHAP result for the worst answer of ruOpenBookQA task

Figures 22 and 23 are visualizing SHAP (SHapley Additive exPlanations) analysis results for different answers to the same question of the ruOpenBookQA. In Figure 22, 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 23, 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.

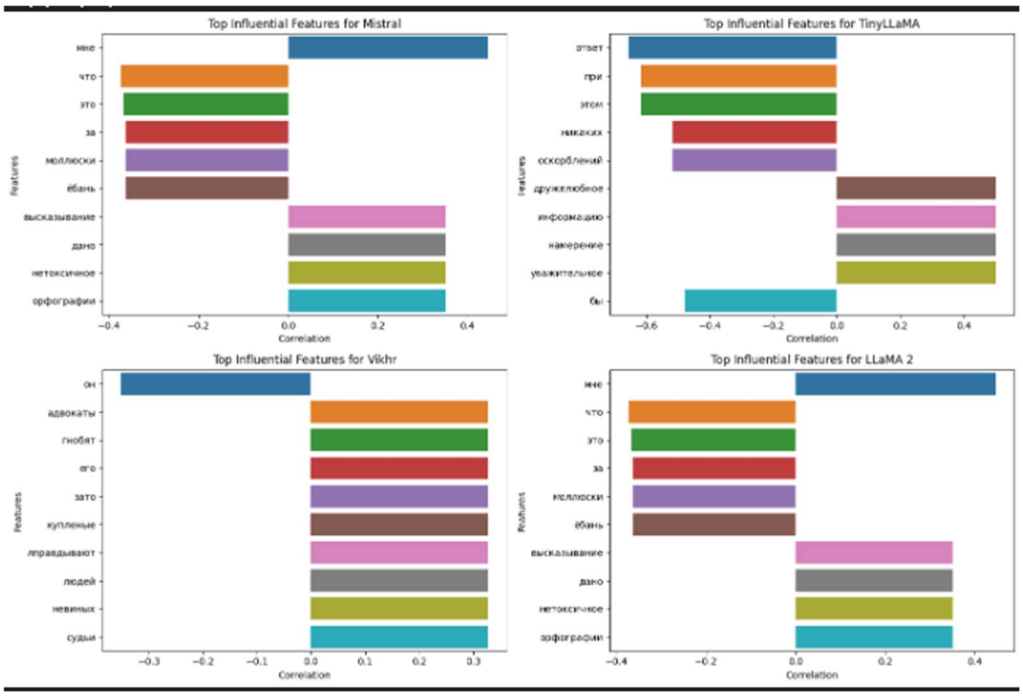

Figure 24. Top LIME influential features for models

This image displays the top influential features for four different language models: Mistral, TinyLLaMA, Vihr, and LLaMA 2. The features are ranked by their correlation coefficients, which indicate the strength and direction of the relationship between the feature and the model's output.

For the Mistral model, the most influential positive features include "мне" "что", "вспомнить", and "рассказывать", while the most influential negative features include "требовать" and "совпадает". The TinyLLaMA model shows that the most positively correlated features are "записи", "определение", and "фраза", while the most negatively correlated features are "выступив" and "возможно".

For the Vihr model, the top influential positive features are "записи", "описание", and "находятся", and the top negative features are "проблема" and "привести". Finally, the LLaMA 2 model has the most positively correlated features as "меня", "может", and "случае", and the most negatively correlated features as "человек" and "был". These insights into the key linguistic and semantic features that drive the performance of these language models can be very useful for understanding their strengths, weaknesses, and potential biases. Analyzing the differences in feature importance across the models can also reveal opportunities for improving model robustness and generalizability.

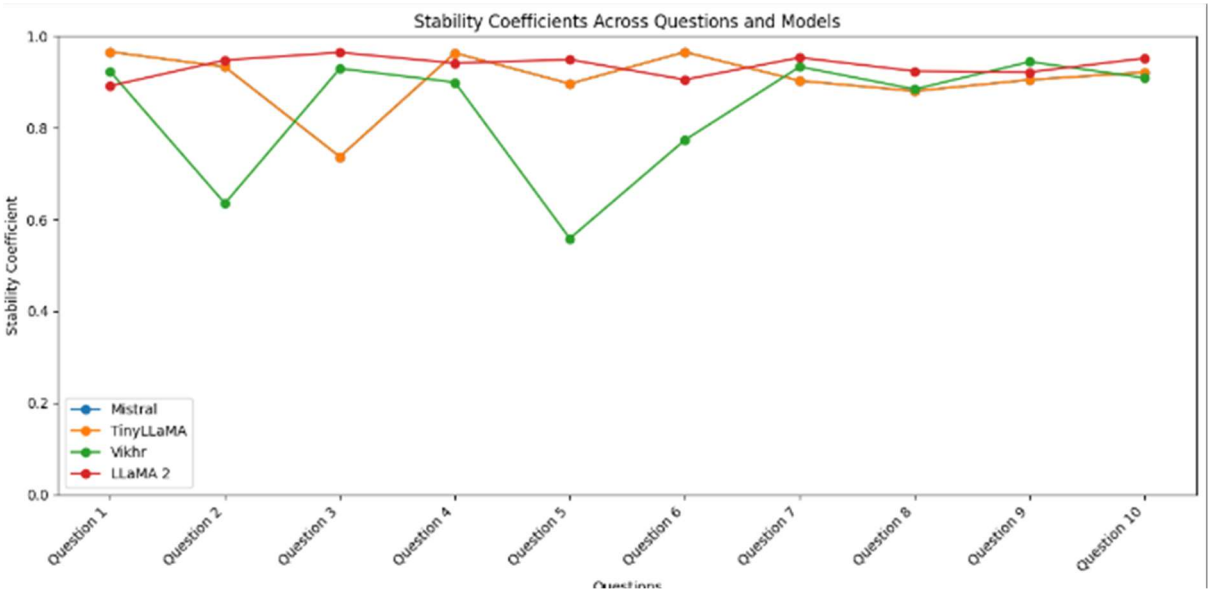

Figure 25. Stability Coefficients Across models

The graph reveals that the stability coefficients for the Mistral, TinyLLaMA, Vihr, and LLaMA 2 models fluctuate across the different questions, with Mistral and LLaMA 2 generally displaying higher stability than TinyLLaMA and Vihr.

For example, Mistral and LLaMA 2 have relatively high stability coefficients for Questions 1, 5, and 9, indicating that their outputs are more consistent for those types of inputs. In contrast, the stability of TinyLLaMA and Vihr appears to be more variable across the different questions.

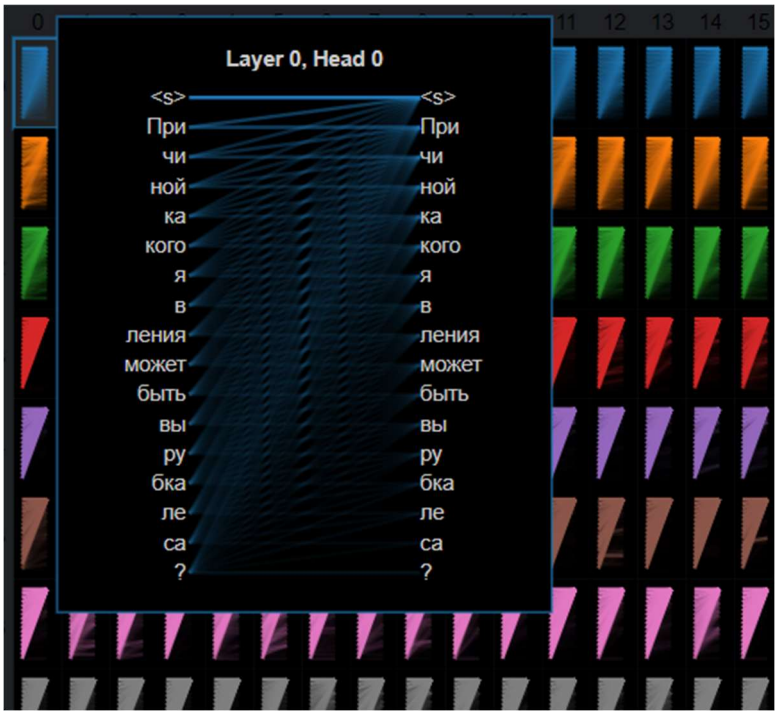

Figure 26. Attention Matrix for the TinyLlama model

In figure 26, we can see that the results of the attention matrix for the ruopenbookQA. This information can be valuable for understanding the reliability and robustness of these language models in different contexts. Models with higher stability are likely to produce more consistent and predictable outputs, which can be important for applications that require dependable and trustworthy natural language processing.

4. 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.