DOI: 10.15514/ISPRAS-2026-38(3)-26

Detection of Human Edits in Russian Scientific Machine-Generated Texts

V.A. Malykh, ORCID: 0000-0002-4508-2527 <val@maly.hk>
M. Dorosh, ORCID: 0009-0002-8154-2495 <doroschmih@yandex.ru>
ITMO University,
bldg. A, 49, Kronverksky Prospekt, St. Petersburg, 197101, Russia.

Abstract. Large language models (LLMs) are rapidly evolving and increasingly integrated into various aspects of life. The texts generated by these models are becoming increasingly indistinguishable from those written by humans, posing significant challenges in identifying synthetic content. In this work, we explore methods for detecting human edits and corrections in abstracts of scientific papers written in Russian and originally generated by various LLMs. In addition to building a strong encoder-based detection model everaging BERT- and RoBERTa-based architectures with current state-of-the-art techniques, we also focus on analysis of robustness to domain shift, aiming for generalization to LLMs not seen during training. We demonstrate that our approach outperforms LLM few-shot learning baselines even on small datasets, and we investigate in which scenarios the addition of a CRF layer improves metrics and in which it does not.

Keywords: large language models; AI content detection; domain generalization.

For citation: Malykh V.A., Dorosh M. Detection of Human Edits in Russian Scientific Machine-Generated Texts. Trudy ISP RAN/Proc. ISP RAS, vol. 38, issue 3, part 2, 2026, pp. 149-160. DOI: 10.15514/ISPRAS-2026-38(3)-26.

Acknowledgements. The authors would like to thank the organizers of the competition held within the Artificial Intelligence and Natural Language (AINL) conference for providing the dataset: Tatiana Batura (Ershov Institute of Informatics Systems SB RAS), Elena Bruches (Ershov Institute of Informatics Systems SB RAS and Novosibirsk State University), and Milana Shvenk (Novosibirsk State University).

Обнаружение человеческих правок в русскоязычных сгенерированных научных текстах

В.А. Малых, ORCID: 0000-0002-4508-2527 <val@maly.hk>
М. Дорош, ORCID: 0009-0002-8154-2495 <doroschmih@yandex.ru>
Университет ИТМО,
Россия, 197101, Санкт-Петербург, Кронверкский проспект, д. 49, корп. А.

Аннотация. Большие языковые модели (LLMs) быстро развиваются и всё активнее внедряются в различные сферы жизни. Тексты, создаваемые этими моделями, становятся всё менее отличимыми от написанных человеком, что создаёт серьёзные трудности при выявлении синтетического контента. В данной работе мы исследуем методы обнаружения человеческих правок и корректировок в аннотациях научных статей на русском языке, изначально сгенерированных различными LLM. Помимо построения мощной модели детектирования на основе энкодеров, использующей архитектуры BERT и RoBERTa с современными методами обучения, мы также сосредоточены на анализе устойчивости к смещению домена, стремясь к обобщению на модели, не встречавшиеся при обучении. Мы показываем, что наш подход превосходит базовые решения на основе LLM в режиме обучения по нескольким примерам даже на небольших выборках, и исследуем, в каких сценариях добавление слоя CRF улучшает метрики, а в каких – нет.

Ключевые слова: Большие языковые модели; детекция сгенерированного контента; обобщение на неизведанные домены.

Для цитирования: Малых В.А., Дорош М. Обнаружение человеческих правок в русскоязычных сгенерированных научных текстах. Труды ИСП РАН, том 38, вып. 3, часть 2, 2026 г., стр. 149–160 (на английском языке). DOI: 10.15514/ISPRAS–2026–38(3)–26.

Благодарности. Авторы выражают благодарность организаторам соревнования в рамках конференции “Artificial Intelligence and Natural Language” (AINL) за предоставленные данные: Татьяне Батуре (Институт систем информатики им. А.П. Ершова СО РАН), Елене Бручес (Институт систем информатики им. А.П. Ершова СО РАН, Новосибирский государственный университет) и Милане Швенк (Новосибирский государственный университет).

1. Introduction

In recent years, the line between machine-generated and human-written content has become increasingly blurred. Two years ago, the online game “Human or Not?” [1], inspired by the Turing test, invited players to chat briefly with either a real human or a large language model (LLM). The results were surprising: across more than 1.5 million sessions, users were only able to correctly identify their conversation partner 60–70% of the time. This accuracy, only slightly better than random guessing, highlighted how far LLMs had come in mimicking human behavior. Given the rapid progress of LLMs over the past two years, it is reasonable to assume that the distinction is now even harder to make, with performance likely dropping closer to the 50% mark – essentially indistinguishable from chance.

Most prior work has therefore treated the problem as a binary classification task: given a text, determine whether it was produced by a human or by a machine. While this framing is natural, it overlooks an increasingly common scenario in practice: many texts are now the product of human–AI collaboration, where an initial draft is generated by an LLM and subsequently refined by a human editor. Detecting such interventions is crucial in domains like academic publishing, journalism and legal writing where distinguishing between raw machine output and human-curated text carries important practical and ethical implications.

In this work, we take a different perspective. Rather than asking whether a text is machine- or human-authored, we assume that the entire text was initially generated by an LLM and focus on identifying which individual words or phrases have been modified by a human. This reformulates

the problem as a token-level classification task: given a fully AI-generated text that may have been edited, predict for each token whether it was altered by a human editor. This setup is closer to well-studied problems such as Named Entity Recognition (NER) or Part-of-Speech (POS) tagging, but it introduces unique challenges related to subtle semantic and stylistic variations introduced during human editing.

The main contributions of this work are:

- First, we formally define the problem of detecting human edits in LLM-generated text as a token-level classification task;
- We demonstrate that our encoder-based transformer approaches outperform LLM few-shot baselines;
- We analyze the trade-off between effective generalization to unseen edits and higher performance on previously;

By addressing this novel formulation, we aim to contribute to a deeper understanding of how humans interact with machine-generated text and how such interactions can be reliably detected in practice.

2. Related Work

The rapid advancement of large language models has spurred significant interest in methods for distinguishing between human- and machine-generated text. As a result, the problem of AI-generated content detection has attracted considerable attention leading to the development of a wide range of research approaches and open-source tools.

For example, the work [2] demonstrates high classification accuracy for outputs from four different LLMs using the GPTZero method [3]. The authors propose a novel ternary system that includes an "undecided" category for texts that are difficult to attribute to a single source. This new category addresses the increasing sophistication of large language models and the blurring lines between human and machine writing. The research also highlights the crucial role of explainability, showing that detectors must provide clear, understandable reasoning for their decisions to help users interpret the results and build trust.

The authors of [4] investigated the applicability of classical machine learning methods, such as the XGBoost Classifier and SVM, alongside the deep learning model BERT, for the task of detecting AI-generated text. The results showed that BERT was the most effective, achieving an accuracy of 93%. In contrast, the XGBoost Classifier and SVM models achieved lower accuracies of 84% and 81%, respectively. The authors concluded that BERT's superior performance can be explained by its ability to better capture the complex patterns in AI-generated language.

In another direction, the paper [5] explores a statistical approach to distinguish between human and machine-authored texts based on distributional properties. The authors proposed MMD-MP - new method to address the instability of existing detectors when trained on texts from various large language models. Their approach modifies the optimization of the Maximum Mean Discrepancy to be "multi-population aware", which allows it to maintain a low variance even when the input texts are stylistically diverse. The results of their experiments showed that MMD-MP achieved superior detection performance and demonstrated better transferability, which means that it could effectively detect texts from LLMs it was not specifically trained on.

The present work [6] proposes a new method called SeqXGPT for sentence-level AI-generated text (AIGT) detection: a task they introduce to address the limitations of existing document-level detectors. Their approach utilizes log probability lists from "white-box" large language models as features, which are then processed by a model based on convolution and self-attention networks. Their experimental results show that while previous methods struggle with sentence-level detection, their proposed method significantly surpasses them and also performs well at the document level.

More recent work [7], formulates the task as a multi-class classification problem. Here the goal is not only to detect whether the code was generated by a human or a machine, but also to identify the specific language model that produced the code in one of several programming languages (e.g., C++, Python, Java). Such approaches emphasize the need for fine-grained classification in practical settings, especially as generated content becomes more sophisticated.

Although the paper [8] does not address the task of detecting human edits in AI-generated texts, it is technically related. The authors proposed a new approach to Portuguese Named Entity Recognition (NER) by training their own Portuguese BERT models and employing a BERT-CRF architecture. They explored both feature-based and fine-tuning training strategies. Their fine-tuning approach achieved new state-of-the-art results on the HAREM I dataset, improving the F1-score by 1 point in the selective scenario (5 entity classes) and by 4 points in the total scenario (10 entity classes). The study concluded that their proposed BERT-CRF model outperforms previous state-of-the-art methods even though it was pre-trained on significantly less data.

The authors of work [9] investigate methods for reliably distinguishing AI-generated text from human-written text. They combine theoretical analysis with empirical evaluations, deriving sample complexity bounds that quantify the amount of text needed for accurate detection. Using multiple datasets (XSum, SQuAD, IMDb, Kaggle FakeNews) and testing various LLMs (GPT-2, GPT-3.5-Turbo, LLaMA, LLaMA-2-13B-Chat-HF, LLaMA-2-70B-Chat-HF) against state-of-the-art detectors (oBERTa-Large/Base-Detector, GPTZero), they demonstrate that detection is feasible and provide formal justification for patterns observed in prior studies, such as the relationship between sequence length and detectability.

Finally, in a comprehensive study [10], the authors evaluated twelve publicly available and two commercial AI text detection tools to assess their reliability and accuracy. They created a unique dataset comprising five categories of documents including original human-written texts, AI-generated texts (as well as texts that were machine-translated), manually edited and machine-paraphrased. The study concluded that the existing detection tools are neither accurate nor reliable, with a tendency to classify most outputs as human-written. Furthermore, techniques such as manual editing or machine paraphrasing were found to substantially degrade the performance of these tools. The results indicated that approximately 20% of AI-generated texts would be misclassified as human-written, rising to around 50% for obfuscated texts.

3. Dataset

In this section, we analyze and explore the original data, describe the process of dataset construction, and highlight its limitations and potential weaknesses that should be taken into account.

3.1 Original Dataset Analysis

The original dataset is divided into training and test subsets, each comprising scientific articles from 10 distinct thematic groups. The training set includes 11,006 articles, while the test set contains 1,234. Each entry contains the article title, a set of keywords, and the original human-written abstract provided by the article's author. For each article, multiple machine-generated abstracts were created based on title and key words using different large language models with the same prompt in all cases. The models were selected to cover a diverse range of scenarios, including varying model scales (from medium to large parameter sizes), licensing frameworks (open-source vs. proprietary), and architectural innovations, ensuring a comprehensive evaluation of the current state-of-the-art in language models:

- GPT-4 Turbo [11] extends the context window to 128K tokens and reduces token costs compared to earlier GPT-4 models. Architecturally similar to GPT-4, it includes optimizations for speed and efficiency. It supports multimodal inputs and retrieval augmentation, with main innovations in scaling context length and throughput while lowering inference cost.

- Gemma 2 (27B) [12] is an efficient transformer model featuring innovations like alternating local/global attention, RMS normalization, logit soft-capping, and Grouped-Query Attention (GQA) to speed inference. It delivers state-of-the-art performance for its size, running at full precision on a single accelerator with high throughput.
- Llama 3.3 (70B) [13] is a dense decoder-only transformer with no MoE, trained on tens of trillions of tokens. It supports eight languages and a 128K token context window. It offers competitive performance with GPT-3.5-class models and focuses on stability and long-context reasoning. All weights are openly released.
- GigaChat [14] uses a sparse Mixture-of-Experts (MoE) transformer with 20B parameters, activating about 3.3B per token. Designed for Russian language tasks, it supports pretrained and DPO-fine-tuned versions. The family is open-source and achieves strong results on Russian and English benchmarks.
- DeepSeek V3 [15] is a 671B-parameter sparse MoE model (37B active) employing novel DeepSeekMoE layers and Multi-Head Latent Attention for efficient inference. Trained on 14.8T tokens with 128K context, it uses Group-Relative Policy Optimization (GRPO) in RLHF for efficient fine-tuning, achieving state-of-the-art open model results with modest compute.

It is important to note that the training and test sets differ significantly. While they share eight scientific topics in common, each subset also includes two unique topics not present in the other. As shown in Fig. 1, the 'oil & gas' and 'physics' topics are present in the training set but absent from the test set. Furthermore, the test set includes an additional machine-generated abstract produced by the DeepSeek [16] model, which is not present in the training set (Table 1). This distinction introduces a domain shift that poses a challenge for model generalization.

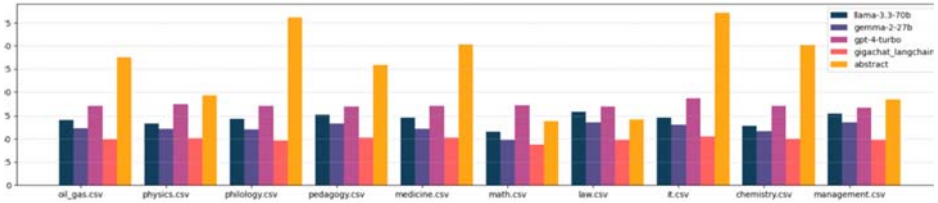

Fig. 1. Distribution of abstract lengths in different topics.

Table 1. Number of articles by LLMs.

Model	Train	Test
GPT-4 Turbo	11006	1234
Gemma2-27B	11006	1234
LLaMA3.3-70B	11006	1234
GigaChat	11006	1234
DeepSeek	0	1234

3.2 Data Processing Methodology and Learning Objective

We model the task as a token-level classification problem: given an AI-generated sentence that may have been partially edited by a human, the goal is to identify which tokens have been modified. For building the dataset, we assume access to aligned pairs of sentences – the original machine-generated version and its potentially human-edited counterpart. For each such pair we compute a binary token-level mask using the change_mask function, the pseudocode for which is shown in Fig. 2.

We use Python's difflib.SequenceMatcher to align the original and edited sentences at the word level and extract edit operations (equal, replace, insert, delete), which we map to binary labels.

Since we ultimately feed the edited sentence into a transformer-based model using a subword tokenizer, special care must be taken to align word-level labels with subword tokens. We therefore propagate each word-level label to all corresponding subword tokens, and assign -100 to special tokens and padding tokens, ensuring that they are ignored during the loss computation in training.

This approach allows for a precise and fine-grained supervision signal, enabling models to learn to detect subtle human edits in machine-generated text.

Generates word-level edit mask — pseudocode

```
function change_mask(original, edited):
    original_words = split original into words
    edited_words = split edited into words

    matcher = SequenceMatcher(original_words, edited_words)
    mask = empty list

    for each (tag, i1, i2, j1, j2) in matcher.get_opcodes():
        if tag == 'equal':
            append 0 to mask (j2 - j1) times
        else:
            append 1 to mask (j2 - j1) times

    return mask
```

Fig. 2. Word-level edit mask generation pseudocode.

3.3 Resulting Dataset Analysis

As a result, the final dataset consists of pairs of modified sentences and corresponding binary masks indicating which tokens were edited. From a machine learning perspective, the task poses several technical challenges. The first and most significant issue is the extreme class imbalance: the figure below show that edited tokens account for only about 1% of all tokens in a typical sentence (Fig. 3). This imbalance makes it difficult for models to learn meaningful signals without resorting to trivial predictions.

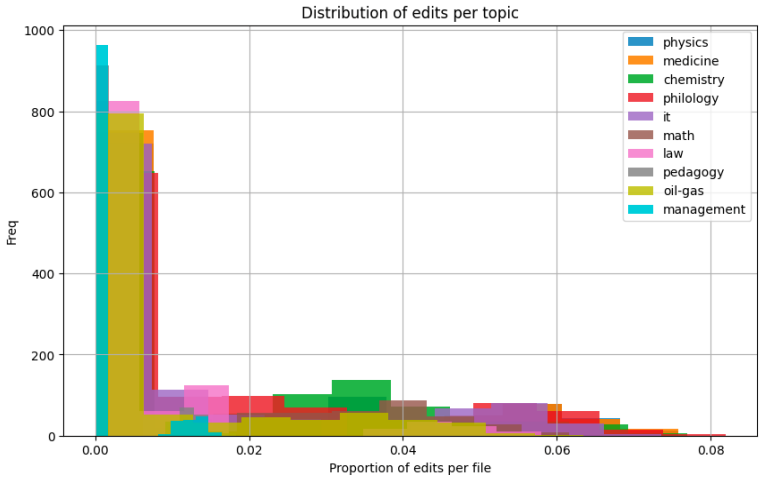

Fig. 3. Distribution of edits per topic.

The second challenge stems from a pronounced domain shift across the data. Although the training and test sets are generally similar in structure and style the test data includes scientific articles on new and previously unseen topics. This introduces a semantic gap between training and test distributions making it harder for models to generalize. In addition, the test set contains examples related to a new model called DeepSeek, which was not present in the training data. This is most evident in the histograms of the most frequent edits (see Fig. 4, 5, and 6).

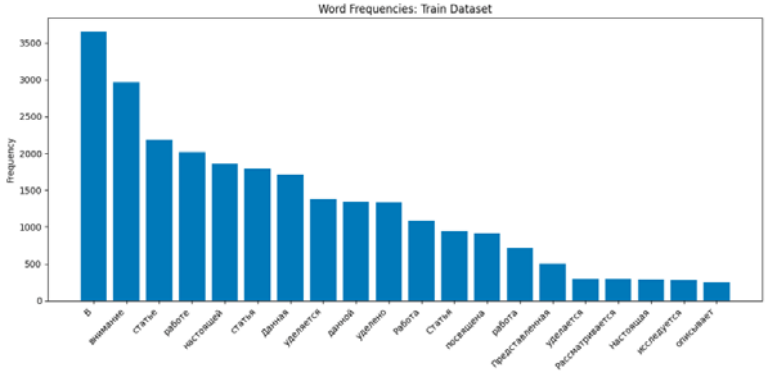

Fig. 4. Distribution of the most frequent edits on train set.

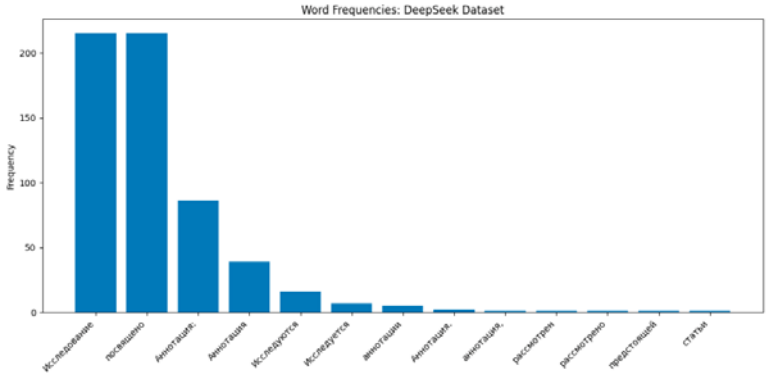

Fig. 5. Distribution of the most frequent edits on test set.

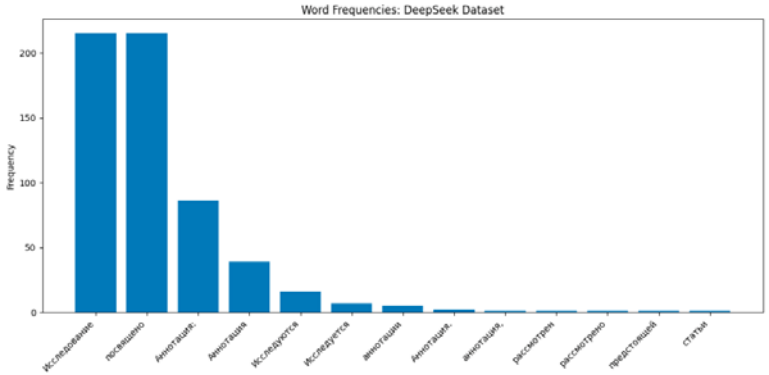

Fig. 6. Distribution of the most frequent edits: DeepSeek.

As a result, the model must not only detect edits in familiar contexts, but also handle unfamiliar terminology, structures, and linguistic patterns introduced by both the new topics and the unseen model. This domain shift increases the difficulty of the task and requires the model to rely on more than simple pattern memorization, emphasizing the importance of robust contextual understanding.

4. Experiments

In this section, we compare several models and approaches for the edit detection task, treating it as a token-level classification problem. We evaluate each method under different scenarios to measure how well the models generalize. These scenarios include in-domain testing, cross-domain testing on new topics and a focused test on data generated by the unseen DeepSeek model. By examining performance across these settings, we aim to highlight the strengths and weaknesses of each approach when faced with shifts in data distribution.

4.1 Baseline

As a baseline, we selected Gemini Flash 2.0, motivated by its favorable performance-to-cost ratio per token. The model was tasked with identifying which words had been replaced in a given lowercase text, with the replacements required to be output in uppercase. To improve performance, we used a few-shot learning setup, providing the model with three example pairs consisting of an original paragraph and its corresponding formatted paragraph, where replaced words were highlighted in uppercase. Few-shot learning is a widely used and practical technique that enables models to quickly adapt to new tasks by showing only a small number of examples, making it particularly effective when large-scale task-specific training data is not available.

4.2 Models

We evaluated three transformer-based models with attention mechanisms [17], all pre-trained on large Russian text corpora. These models are DeepPavlov/rubert-base-cased, ai-forever/ruBert-large, and sberbank-ai/ruRoberta- large. Each of them is based on the BERT architecture and has been widely used in various natural language processing tasks in Russian. We used these models as encoders to process input sequences and extract contextualized token representations.

To further improve the detection of edit boundaries in the best-performing model, we added a Conditional Random Fields (CRF) [18] layer on top (Fig. 7).

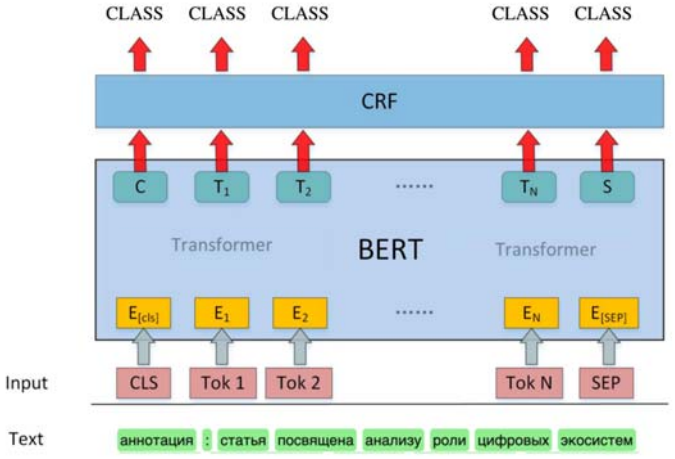

Fig. 7. Model overview.

CRF is a structured prediction algorithm often used in sequence labeling tasks, such as named entity recognition [19, 8] or part-of-speech tagging. It allows the model to take into account dependencies between neighboring labels that helps in producing more consistent and accurate label sequences, especially near the edges of changes or edits.

Formally, the score of a label sequence $y = (y_1, \dots, y_T)$ (binary in our case) given input x is defined as:

$$\text{score}(x,y) = \sum_{t=1}^{T} \left(s(x,t)_{y_t} + A_{y_{t-1},y_t} \right) \quad (1)$$

where

- $s(x,t)_{y_t} = Wh_t + b$ is the emission score from BERT for token t and label y_t obtained by applying Linear layer over the last layer embeddings
- A_{y_{t-1},y_t} is the learned transition score from label $y_{t-1}$to y_t

The probability of a label sequence y is given by:

$$P(y \mid x) = \frac{\exp(\text{score}(x,y))}{\sum_{y'} \exp(\text{score}(x,y'))} \quad (2)$$

The CRF layer is trained on top of a frozen transformer encoder, the architecture of which is shown in Fig. 7, adding only a small number of additional parameters while resulting in a noticeable improvement in boundary-level accuracy. By freezing the transformer, we ensure that only the CRF transition matrix and the final linear layer are learned. This addition helps the model better capture the structure of label transitions and improves boundary detection performance.

4.3 Setup

In all experiments, no hyperparameter tuning was performed, and the same fixed configuration was used throughout (Table 2). Specifically, the entire transformer encoder was frozen, and only the final linear classification layer was trained, except for the experiment that included a CRF layer on top.

All experiments were run under these conditions to ensure that performance differences reflected architectural choices rather than hyperparameter optimization.

Table 2. Training configuration details.

Hyperparameter	Value
Optimizer	Adam [20]
Learning rate	5.0e-5
Loss function	Weighted cross-entropy loss [21]
Class weight ratio (pos:neg)	0.1
Dropout rate	0.2
Batch size	32

5. Results

We used the F1-score as the main evaluation metric due to class imbalance in the dataset. In our task the number of tokens labeled as non-edits significantly exceeds the number of edited tokens. Accuracy would not reflect true performance because a model could achieve high accuracy by predicting only the majority class. The F1-score provides a more reliable measure of the model's ability to detect edits under these conditions. The results and an example are presented in Table 3 and Fig. 8 respectively.

Table 3. Model performance.

Model	CRF	F1-score (train) (%)	F1-score (test) (%)	F1-score (DeepSeek) (%)
Gemini 2.0 Flash (Baseline)	-	-	85.4	76.4
DeepPavlov/rubert-base-cased	-	84.4	74.4	55.0
ai-forever/ruBert-large	-	97.0	91.5	76.3
sberbank-ai/ruRoberta-large	-	97.2	93.9	**90.8**
ai-forever/ruBert-large	+	96.7	97.9	51.2
sberbank-ai/ruRoberta-large	+	96.9	**98.2**	53.3

Fig. 8. An example of classification on a text from the test set using sberbank-ai/ruRoberta-large with CRF.

5.1 Baseline Comparison

Despite the lack of direct fine-tuning on the data the few-shot learning approach demonstrated strong performance, outperforming DeepPavlov/rubert-base-cased. Furthermore, this approach proved robust to new data generated by the DeepSeek model, surpassing ai-forever/ruBert-large. However, sberbank-ai/ruRoberta-large still achieved the best results on both the test set and the DeepSeek subset. It is also worth noting that, despite the apparent simplicity of the few-shot learning approach and the absence of fine-tuning, the response generation time via Google API is comparable to the training time of an encoder-based model, which provides higher overall performance. Based on this, the comparison of our approach with the proposed baseline can be considered relevant and fair.

5.2 Test set without domain shift

The best overall performance on the full test set was achieved by the sberbank-ai/ruRoberta-large model with an additional CRF layer. This combination proved especially effective at capturing label dependencies and accurately detecting edit spans. However, its performance drops noticeably on a small subset of test data generated by the DeepSeek model, which was not seen during training. This suggests that while the model is highly optimized for familiar patterns and training-like inputs, it struggles to generalize to unseen distributions. The CRF layer, which helps enforce consistent label sequences, may also contribute to overfitting on known token transitions, making it less flexible when encountering new linguistic structures or vocabulary.

5.3 New model (DeepSeek)

In contrast, a simpler model without a CRF layer shows better robustness to domain shift and performs more reliably on the DeepSeek subset. This highlights a trade-off between sequence-level precision on in-domain data and generalization to novel inputs. Regarding the new topics that were unseen during initial LLM training, the model still generalizes well and achieves performance comparable to its metric scores on topics overlapping with the training data. The results are presented in Table 4.

5.4 Generalization with Limited Data

A high F1-score achieved by the model indicates its robustness to limited annotated data and imbalanced label distributions. Despite the small size of the dataset and the low proportion of edited tokens, the model is able to detect edits reliably without overfitting. This suggests that the combination of pre-trained language representations and appropriate fine-tuning allows the model to generalize well even under resource-constrained conditions.

Table 4. Distribution of code snippets across topics in train and test splits.

Topic	**F1-score on Test set**
Philology	0.989
Law	0.996
Medicine	0.978
Pedagogy	0.996
IT	0.943
Chemistry	0.97
Physics	0.99
Math	0.938
Biology	**0.973**
Economics	**0.99**

6. Conclusion

n this paper, we compared several deep learning approaches for the task of edit detection, formulating it as a token-level classification problem. We evaluated multiple transformer-based models pre-trained on Russian, including ruBERT [22] and ruRoberta [23] and explored the effect of adding a Conditional Random Fields (CRF) layer for improved sequence labeling. Our experiments covered various scenarios, including in-domain testing, cross-topic generalization and a domain shift setting involving previously unseen data generated by the DeepSeek model. The results highlight both the effectiveness and the limitations of current models when faced with imbalanced labels, limited annotated data and distributional shifts. We compare our approaches with state-of-the-art large language models and demonstrate that they outperform current few-shot learning methods. Overall, we show that strong pre-trained encoder-based models can still achieve high accuracy in realistic, low-resource settings and remain robust in challenging conditions.

References

[1]. Jannai D., Meron A., Lenz B., Levine Y., Shoham Y. Human or Not? A Gamified Approach to the Turing Test. 2023. arXiv:2305.20010. Available at: https://arxiv.org/abs/2305.20010, accessed 05.11.2025.

[2]. Ji J., Li R., Li S., Guo J., Qiu W., Huang Zh., Chen Ch., Jiang X., Lu X. Detecting Machine-Generated Texts: Not Just “AI vs Humans” and Explainability is Complicated. 2025. arXiv:2406.18259. Available at: https://arxiv.org/abs/2406.18259, accessed 05.11.2025.

[3]. Emi B., Spero M. Technical Report on the Pangram AI-Generated Text Classifier. 2024. arXiv:2402.14873. Available at: https://arxiv.org/abs/2402.14873, accessed 05.11.2025.

[4]. Prova N. Detecting AI Generated Text Based on NLP and Machine Learning Approaches. 2024. arXiv:2404.10032. Available at: https://arxiv.org/abs/2404.10032, accessed 05.11.2025.

[5]. Zhang Sh., Song Y., Yang J., Li Yu., Han B., Tan M. Detecting Machine-Generated Texts by Multi-Population Aware Optimization for Maximum Mean Discrepancy. 2024. arXiv:2402.16041. Available at: https://arxiv.org/abs/2402.16041, accessed 05.11.2025.

[6]. Wang P., Linyang Li, Ren Кю, Jiang B., Zhang D., Qiu X.. SeqXGPT: Sentence-Level AI-Generated Text Detection. 2023. arXiv:2310.08903. Available at: https://arxiv.org/abs/2310.08903, accessed 05.11.2025.

[7]. Orel D., Azizov D., Nakov P. CoDet-M4: Detecting Machine-Generated Code in Multi-Lingual, Multi-Generator and Multi-Domain Settings. 2025. arXiv:2503.13733. Available at: https://arxiv.org/abs/2503.13733, accessed 05.11.2025.

[8]. Souza F., Nogueira R., Lotufo R. Portuguese Named Entity Recognition using BERT-CRF. 2020. arXiv:1909.10649. Available at: https://arxiv.org/abs/1909.10649, accessed 05.11.2025.

[9]. Chakraborty S., Bedi A.S., Zhu S., An B., Manocha D., Huang F. On the Possibilities of AI-Generated Text Detection. 2023. arXiv:2304.04736. Available at: https://arxiv.org/abs/2304.04736, accessed 05.11.2025.

[10]. Weber-Wulff D., Anohina-Naumeca A., Bjelobaba S., et al. Testing of Detection Tools for AI-Generated Text. International Journal of Educational Integrity, 19:26, 2023. DOI: 10.1007/s40979-023-00146-z.

[11]. OpenAI and Achiam J. et al. GPT-4 Technical Report. 2024. arXiv:2303.08774. Available at: https://arxiv.org/abs/2303.08774, accessed 05.11.2025.

[12]. Gemma Team and Riviere M. et al. Gemma 2: Improving Open Language Models at a Practical Size. 2024. arXiv:2408.00118. Available at: https://arxiv.org/abs/2408.00118, accessed 05.11.2025.

[13]. Grattafiori A. et al. The Llama 3 Herd of Models. 2024. arXiv:2407.21783. Available at: https://arxiv.org/abs/2407.21783, accessed 05.11.2025.

[14]. GigaChat team, Mamedov V. et al. GigaChat Family: Efficient Russian Language Modeling Through Mixture of Experts Architecture. 2025. arXiv:2506.09440. Available at: https://arxiv.org/abs/2506.09440, accessed 05.11.2025.

[15]. DeepSeek-AI, Liu A. et al. DeepSeek-V3 Technical Report. 2025. arXiv:2412.19437. Available at: https://arxiv.org/abs/2412.19437.

[16]. DeepSeek-AI, Guo D. et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. 2025. arXiv:2501.12948. Available at: https://arxiv.org/abs/2501.12948, accessed 05.11.2025.

[17]. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I. Attention Is All You Need. 2023. arXiv:1706.03762. URL: https://arxiv.org/abs/1706.03762, accessed 05.11.2025.

[18]. Sutton Ch., McCallum A. An Introduction to Conditional Random Fields. 2010. arXiv:1011.4088. Available at: https://arxiv.org/abs/1011.4088, accessed 05.11.2025.

[19]. Huang W., Cheng X., Wang T., Chu W. BERT-Based Multi-Head Selection for Joint Entity-Relation Extraction. 2019. arXiv:1908.05908. Available at: https://arxiv.org/abs/1908.05908, accessed 05.11.2025.

[20]. Kingma D.P., Ba J. Adam: A Method for Stochastic Optimization. 2017. arXiv:1412.6980. Available at: https://arxiv.org/abs/1412.6980, accessed 05.11.2025.

[21]. Mao A., Mohri M., Zhong Yu. Cross-Entropy Loss Functions: Theoretical Analysis and Applications. 2023. arXiv:2304.07288. Available at: https://arxiv.org/abs/2304.07288, accessed 05.11.2025.

[22]. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. arXiv:1810.04805. Available at: https://arxiv.org/abs/1810.04805, accessed 05.11.2025.

[23]. Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer M., Stoyanov V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019. arXiv:1907.11692. Available at: https://arxiv.org/abs/1907.11692, accessed 05.11.2025.

Информация об авторах / Information about authors

Дорош Мишель – аспирант университета ИТМО. Научные интересы: обработка естественного языка, большие языковые модели, глубокое обучение, временные ряды.

Dorosh Mishel – postgraduate student at ITMO University. Research interests include natural language processing, large language models, deep learning, and time series analysis.

Валентин Андреевич Малых – кандидат технических наук, руководитель исследований в области обработки естественного языка в компании MTS AI, сотрудничающий с ИТМО. Его научные интересы включают генерацию естественного языка, глубокое обучение и применение ИИ в компьютерном зрении и медицинской визуализации.

Valentin Andreevich Malykh – Cand. Sci. (Tech.), Prof., Head of NLP Research at MTS AI, collaborating with ITMO University. His research interests include natural language generation, deep learning, and the application of AI to computer vision and medical imaging.