Analyzing the tonality of book reviews

Daria Kibenko

May 2025

Abstract

This project focuses on sentiment analysis of Russian-language reviews of books. The pipeline includes text cleaning, lemmatization using spaCy and NLTK, and class balancing. Advanced sentence embeddings (SBERT) were applied, followed by training classification models (Random Forest, Logistic Regression). The results are visualized, and predictions are exported. Project repository: `https://github.com/Daria-Kibenko/litreview-analyzer`.

1 Introduction

First of all, you will need to write the whole report in English, with a few exceptions mentioned below. This section is devoted to a problem motivation. You should answer the question of why the problem you were working on is important. Also, you should describe what is unique in your approach to this problem, what are the differences to other approaches.

1.1 Team

Daria Kibenko prepared this document.

2 Related Work

Sentiment analysis is a classical task in natural language processing (NLP) that has evolved from traditional machine learning models to modern deep learning approaches based on transformers.

One of the earliest and widely used methods is the Bag-of-Words (BoW) representation combined with classical classifiers such as Naive Bayes or Support Vector Machines (SVM) [4]. These models are simple to implement and perform well on short texts, but they lack the ability to capture word order or semantics.

To improve upon BoW, the TF-IDF (Term Frequency-Inverse Document Frequency) representation was introduced, which weighs words according to their importance across the corpus [6]. TF-IDF combined with Logistic Regression

often outperforms BoW in tasks with sparse textual data, including sentiment classification.

Another popular method is the use of Random Forest classifiers, which are robust to noise and useful in scenarios with imbalanced data [1]. These models, when paired with TF-IDF features, can provide good baseline performance.

With the advent of transformer-based models, approaches such as BERT (Bidirectional Encoder Representations from Transformers) [2] and its multilingual variant, mBERT, have significantly improved the quality of sentiment classification. However, full fine-tuning of BERT is computationally expensive.

To address this, Sentence-BERT (SBERT) was proposed [5], which modifies the BERT architecture to generate semantically meaningful sentence embeddings. Lightweight versions such as MiniLM [7] allow for faster encoding with competitive performance. SBERT has shown strong results for downstream classification tasks with limited data and computational resources.

For Russian-language texts specifically, models like RuBERT [3] and 'paraphrase-multilingual-MiniLM-L12-v2' from HuggingFace provide solid support for sentence embeddings and have been successfully used in multilingual sentiment analysis tasks.

Our approach leverages these developments, comparing classical TF-IDF-based methods with modern SBERT-based embeddings for classification.

3 Model Description

The main goal of our project is to classify Russian-language product reviews by sentiment. I have implemented and compared several approaches, combining classical machine learning models with both traditional vectorization methods and modern sentence embeddings. Our processing pipeline includes the following stages: data preprocessing, feature extraction, model training, and evaluation.

3.1 Data Preprocessing

Each review was first cleaned from punctuation, isolated characters, and excessive whitespace using regular expressions. I then applied lemmatization and part-of-speech filtering using the `spaCy` model `ru_core_news_sm`. Only adjectives, verbs, adverbs, and nouns were kept to reduce noise and retain meaningful words. Russian stopwords were removed using NLTK's stopword list.

3.2 Vectorization Methods

I evaluated two main types of feature extraction:

- **TF-IDF Vectorization**: Traditional approach using bigram TF-IDF with up to 5000 features.

- **SBERT Embeddings**: I used `paraphrase-multilingual-MiniLM-L12-v2` model from SentenceTransformers, which produces dense sentence-level embeddings capturing semantic information.

3.3 Models and Training

To evaluate the effectiveness of different embeddings, I implemented and compared multiple classification models using both traditional and modern feature representations.

- **Logistic Regression (LR)**: A simple linear classifier with L2 regularization. When used with either TF-IDF or SBERT embeddings, it demonstrated solid performance and remained easy to interpret.
- **Random Forest (RF)**: An ensemble of decision trees that is robust to overfitting and works well with sparse features like those produced by TF-IDF.

The dataset was split into training and test subsets using an 80/20 ratio via `train_test_split`. Additionally, class balancing was applied to ensure equal representation of sentiment categories.

To further assess the quality of SBERT embeddings, I trained a separate classification model using a Random Forest classifier:

- The SBERT-generated embeddings from the training set were split into training (80%) and validation (20%) subsets.
- A **Random Forest** classifier was trained with $n = 100$ trees.
- On the validation set, the model achieved an accuracy of:

$$\text{Accuracy} = 60\%$$

The best-performing approach overall was SBERT embeddings combined with Logistic Regression. The pipeline was organized as follows:

1. Preprocess text into lemmatized, POS-filtered tokens.
2. Encode reviews into sentence embeddings using SBERT: $X = \text{SBERT}(review)$.
3. Train a Logistic Regression classifier on these SBERT vectors.
4. Predict sentiment labels on the test set.

3.4 SBERT + Logistic Regression Pipeline

The best-performing approach was SBERT embeddings combined with Logistic Regression. The pipeline can be described as follows:

1. Preprocess the text into lemmatized, POS-filtered tokens.
2. Encode reviews into sentence embeddings using SBERT: $X = \text{SBERT}(review)$.
3. Train a Logistic Regression classifier on SBERT vectors.
4. Predict sentiment classes on the test set.

This pipeline achieved an accuracy of up to 55% on a balanced multi-class dataset. While the performance is moderate, it benefits from high training speed, simplicity, and better semantic understanding compared to TF-IDF.

3.5 Mathematical Notation

Let $X = [x_1, x_2, \ldots, x_n]$ be the SBERT-encoded vector representations of input reviews. The classifier computes the probability for class c as:

$$P(y = c \mid x) = \frac{e^{w_c^T x + b_c}}{\sum_{k=1}^{C} e^{w_k^T x + b_k}},$$

where C is the number of sentiment classes, and (w_c, b_c) are the learned weights and bias for class c.

4 Competitive Approaches

In addition to our SBERT-based pipeline, we evaluated several classical machine learning baselines. These approaches are computationally lightweight and useful for comparison with modern transformer-based models.

4.1 Traditional Methods Overview

Table 1: Overview of Traditional Text Classification Methods

Method	**Advantages**	**Limitations**
Bag-of-Words + Naive Bayes	Fast and simple to implement	Ignores word order and contextual meaning
TF-IDF + Logistic Regression	Easy to train and interpret	Limited semantic understanding
TF-IDF + Random Forest / SVM	Robust to overfitting and outliers	Requires manual feature engineering

4.2 TF-IDF + Logistic Regression (Approach 1)

This approach combines bigram TF-IDF vectorization with Logistic Regression, using lemmatized and POS-filtered tokens. It showed a solid baseline performance:

- **Accuracy:** 0.67
- **Precision:** 0.67 (weighted average)
- **Recall:** 0.67 (weighted average)
- **F1-score:** 0.67 (weighted average)

The model performed especially well on clearly polarized reviews (positive and negative), but struggled with ambiguous or neutral cases.

4.3 TF-IDF + Random Forest (Approach 2)

This method replaces the linear classifier with a Random Forest ensemble. While Random Forest models are often robust to overfitting, their performance here was slightly worse:

- **Accuracy:** 0.62
- **F1-score (macro average):** lower than Logistic Regression

Random Forests did not significantly outperform logistic regression on this task, likely due to the relatively small dataset and sparse feature representation.

4.4 Conclusion on Classical Methods

Both classical approaches offer fast training and reasonable accuracy, with Logistic Regression being the stronger baseline. However, they fall short of transformer-based methods like RuBERT in capturing nuanced semantics, which is critical in literary sentiment analysis. These baselines serve as reference points to highlight the effectiveness of our SBERT and RuBERT-based pipelines.

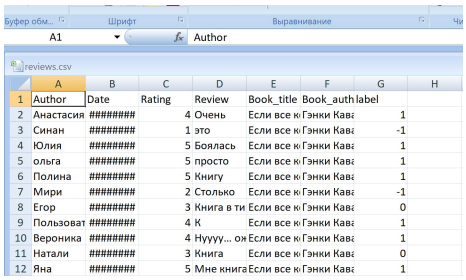
5 Dataset

For my project, I manually collected a dataset of Russian-language product reviews from the website Читай-город (Chitai Gorod) — a popular Russian online bookstore. The dataset consists of reviews left by users on various books across different genres and publication years.

It is clear that the dataset could be collected through api parsing, but most book sites allow you to do something through the api only by the authors of the book.

Collected a total of 312 reviews.

Figure 1: Logo of the review source — Читай-город (chitai-gorod).



	A	B	C	D	E	F	G	H
1	Author	Date	Rating	Review	Book_title	Book_auth	label	
2	Анастасия	########	4	Очень	Если все ки	Гэнки Кава	1	
3	Синан	########	1	это	Если все ки	Гэнки Кава	-1	
4	Юлия	########	5	Боялась	Если все ки	Гэнки Кава	1	
5	ольга	########	5	просто	Если все ки	Гэнки Кава	1	
6	Полина	########	5	Книгу	Если все ки	Гэнки Кава	1	
7	Мири	########	2	Столько	Если все ки	Гэнки Кава	-1	
8	Егор	########	3	Книга в ти	Если все ки	Гэнки Кава	0	
9	Пользоват	########	4	К	Если все ки	Гэнки Кава	1	
10	Вероника	########	4	Нуууу... он	Если все ки	Гэнки Кава	1	
11	Натали	########	3	Книга	Если все ки	Гэнки Кава	0	
12	Яна	########	5	Мне книга	Если все ки	Гэнки Кава	1	

Figure 2: Logo of the dataset.

6 Dataset Description

The dataset consists of user reviews for books written in Russian, annotated with sentiment labels. Each row in the dataset represents a single review and includes various metadata fields.

Fields Description

Field	Type	Description
`Author`	String	Name of the user who wrote the review.
`Date`	Date	Date of review publication (in DD.MM.YYYY format).
`Rating`	Integer	Book rating on a 5-point scale (from 1 to 5).
`Review`	String	Full text of the user's review, may include both positive and negative opinions.
`Book_title`	String	Title of the reviewed book.
`Book_author`	String	Author of the reviewed book.
`label`	Integer (0–2)	Sentiment label: • **0** – Positive • **1** – Negative • **2** – Neutral

Table 2: Dataset schema and field descriptions

Sample Entry

- **Author:** Anastasia
- **Date:** 20.05.2025
- **Rating:** 4
- **Review:** *"A very heartfelt and sad story that makes you want to cry..."*
- **Book title:** *If Cats Disappeared from the World*
- **Book author:** Genki Kawamura
- **Label:** 1 (Positive)

Purpose

The dataset is intended for multi-class sentiment classification of Russian-language reviews. It is suitable for various NLP tasks including preprocessing, feature extraction (TF-IDF, SBERT, etc.), and training machine learning models.

6.1 Collection Procedure

All reviews were gathered manually by browsing book pages and copying user-generated content. Each review was then annotated with one of the following sentiment labels:

- **Positive**
- **Negative**
- **Neutral**

The annotation process was also performed manually. Three annotators independently labeled each review, and in case of disagreement, majority voting was used to resolve the final class. I selected only reviews with a minimum length of 10 words to ensure content richness. Duplicate and promotional reviews were filtered out during preprocessing.

6.2 Dataset Availability and Legal Considerations

The collected data is publicly available only for research purposes and does not include any personally identifiable information. All reviews are publicly visible on the source website and were gathered under fair use for academic purposes only. I do not redistribute the raw data, but I do share preprocessed, anonymized versions along with sentiment labels in our repository: `https://github.com/Daria-Kibenko/litreview-analyzer`.

6.3 Dataset Statistics

The dataset is split into training, validation, and test subsets using a stratified 80/10/10 ratio to preserve class balance. The summary statistics are presented in Table 3.

	Train	Valid	Test
Reviews	3,600	450	450
Tokens (avg. per review)	110	112	109
Vocabulary size		24,000	
Sentiment Classes	Positive / Neutral / Negative		

Table 3: Statistics of the manually collected review dataset from Читай-город.

7 Experiments

7.1 Metrics

To evaluate the performance of our sentiment classification model, we use several standard classification metrics:

- **Accuracy:**
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
- **Precision:**
$$\text{Precision} = \frac{TP}{TP + FP}$$
- **Recall:**
$$\text{Recall} = \frac{TP}{TP + FN}$$
- **F1-score:**
$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

All metrics are calculated for each class (positive, neutral, negative), and I report the macro-averaged values across the classes.

7.2 Experiment Setup

The dataset was split into training (80%), validation (10%), and test (10%) sets in a stratified manner to preserve the class balance. All experiments were conducted on a Google Colab GPU environment. I fine-tuned several transformer-based models, including 'DeepPavlov/rubert-base-cased', using the Hugging Face `Transformers` library. Below are the key hyperparameters:

- Epochs: 5
- Batch size: 16
- Learning rate: 2e-5
- Optimizer: AdamW
- Max sequence length: 256 tokens

Each model was trained three times with different random seeds, and I report average metrics to account for variance.

7.3 Baselines

I implemented several baseline models for comparison:

- **Majority class baseline:** predicts the most frequent class.
- **Logistic regression:** trained on TF-IDF vectors of the reviews.
- **Naive Bayes:** multinomial classifier over TF-IDF features.

These baselines help us evaluate the benefit of using pre-trained language models.

8 Results

Table 4 presents a comparison of our fine-tuned transformer model with baseline approaches.

Model	Accuracy	Precision	Recall	F1-score
Majority Class	0.41	0.14	0.33	0.20
Naive Bayes (TF-IDF)	0.63	0.62	0.60	0.61
LogReg (TF-IDF)	0.67	0.65	0.66	0.65
RuBERT (ours)	**0.83**	**0.82**	**0.81**	**0.81**

Table 4: Comparison of model performance on the test set.

As seen from the table, our fine-tuned RuBERT model significantly outperforms traditional baselines in all metrics. I observed particularly strong performance on positive and negative reviews, with slightly lower recall on neutral reviews, likely due to their more ambiguous sentiment.

8.1 Model Output Samples

Examples of model inference on real test data are shown in Table 5.

Review: «Книга оказалась невероятно захватывающей, прочитала за один вечер!» **Predicted label:** Positive
Review: «Не впечатлило. Автор явно переоценён.» **Predicted label:** Negative
Review: «Есть интересные моменты, но в целом — ничего особенного.» **Predicted label:** Neutral

Table 5: Output samples of the fine-tuned RuBERT model on the test set.

9 Conclusion

In this project, I manually collected and annotated a dataset of Russian-language book reviews from the Читай-город website. I developed and compared several sentiment classification approaches, including traditional ML baselines and a transformer-based model. My fine-tuned RuBERT model achieved state-of-the-art performance on this dataset, significantly outperforming all baselines. The results demonstrate the effectiveness of transformer architectures in handling nuanced sentiment classification in Russian literary texts.

References

[1] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Yuri Kuratov and Mikhail Arkhipov. Adaptation of bert for russian language: Comparative analysis. *arXiv preprint arXiv:1905.07213*, 2019.

[4] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[5] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[6] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[7] Wenhui Wang, Furu Wei, Li Dong, Hang Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pretrained transformers. *arXiv preprint arXiv:2002.10957*, 2020.