DOI: 10.55648/1998-6920-2025-19-4-3-16 УДК 004.912+004.853+811.512

Sentiment analysis of Uzbek texts using NER: a comparative study of SVM, LSTM, and BERT models

B. R. Saidov[1], V. B. Barakhnin[1,2]

[1]Novosibirsk State University
[2]Federal Research Center for Information and Computational Technologies

Abstract: This paper presents a comparative analysis of machine learning (SVM), deep learning (LSTM), and transformer-based (BERT) models for sentiment classification in Uzbek texts, enhanced by Named Entity Recognition (NER). The study addresses the challenge of accurately detecting sentiment in morphologically complex languages with limited resources, focusing on Uzbek–a Turkic language with rich agglutinative structures. A dataset of 10,000 user-generated comments from social platforms was annotated using a hybrid approach: manual labeling for sentiment (positive, negative, neutral) and a CRF-based NER system to identify entities (e.g., brands, locations, public figures). The integration of NER features aimed to resolve contextual ambiguities, such as distinguishing between "I love Samarkand’s history" (positive) and "Samarkand’s traffic is unbearable" (negative). Experimental results demonstrate that BERT, fine-tuned on Uzbek text, achieved the highest accuracy (90.2%) by leveraging contextualized embeddings to align entities with sentiment. LSTM showed competitive performance (85.1%) in sequential pattern learning but required extensive training data. SVM, while computationally efficient, lagged at 78.3% accuracy due to its inability to capture nuanced linguistic dependencies. The findings emphasize the critical role of NER in low-resource languages for disambiguating sentiment triggers and propose practical guidelines for deploying BERT in real-world applications, such as customer feedback analysis. Limitations, including data scarcity and computational costs, are discussed to inform future research on optimizing lightweight models for Uzbek NLP tasks.

The research was carried out within the state assignment of Ministry of Science and Higher Education of the Russian Federation for Federal Research Center for Information and Computational Technologies.

Keywords: Sentiment Analysis, Named Entity Recognition, Uzbek Language, BERT, Low-Resource NLP

For citation: Saidov B. R., Barakhnin V. B. Sentiment analysis of Uzbek texts using NER: a comparative study of SVM, LSTM, and BERT models [Paper Preparation Manual for Vestnik SibGUTI]. *Vestnik SibGUTI*, 2025, vol. 19, no. 4, pp. 3-16. https://doi.org/10.55648/1998-6920-2025-19-4-3-16.

The article was submitted: 28.04.2025;
revised version: 20.05.2025;
accepted for publication 30.07.2025.

1. Introduction

In the modern digital world, the volume of textual data is increasing dramatically. In particular, social networks, online comments and customer feedback are collected on a large scale, and their automatic analysis plays an important role in the social sphere, marketing and decision-making processes. The field of sentiment analysis in the Uzbek language is particularly challenging:

the complex adjectival structure of the language, the abundance of morphological changes and the limited resources (e.g., corpora, pre-trained models) make accurate analysis difficult. In such conditions, technologies such as Named Object Recognition (NER) [1] can play a crucial role in understanding the emotional context. For example, in the sentences *Samarqand me'morchiligi hayratlanarli – Samarkand architecture is amazing* and *Samarqand yoʻllari eski – The roads in Samarkand are old*, although the object *Samarkand* is the same, the first case expresses a positive feeling, and the second case expresses a negative one.

To date, work on sentiment analysis in Uzbek has been limited to simple statistical methods (e.g., SVM [2]) or rule-based systems (Kholmirzayev, 2021). However, modern approaches such as deep learning and transform models (BERT [3]) promise more accurate results for complex languages. At the same time, there is almost no research on the application of NER systems integrated with sentiment analysis to Uzbek [4].

The main goal of this study is to compare the effectiveness of different models (SVM, LSTM [5], BERT) in detecting sentiment in Uzbek texts using NER, taking into account the context. The following issues were addressed:

1. What accuracy does a sentiment analysis system integrated with NER provide in Uzbek?
2. Which model (SVM, LSTM, BERT) can be the most optimal solution in resource-limited conditions?
3. How do errors in the NER system affect sentiment analysis?

The scientific novelty of the study is that for the first time in Uzbek, the effectiveness of transformer models (BERT) combining NER and sentiment analysis has been experimentally evaluated. The results contain valuable recommendations not only for the academic community, but also for practical areas (for example, monitoring customer reviews).

The structure of the article is as follows: Section 2 describes the methodology and data set, Section 3 analyzes the experimental results, and Section 4 indicates directions for future work.

2. Main Sections

2.1. Data Collection and Pre-Processing

2.1.1. Data Collection and Annotation

10,000 Uzbek text comments were collected for the study from social media (Twitter, Facebook, Telegram) and e-commerce sites. The data were selected based on the following criteria:

1. Topic diversification: various sectors (restaurants, tourism, technology, culture).
2. Sentiment balance: positive (50%), negative (30%), neutral (20%) – this distribution was adjusted to the ratio naturally observed in social media.
3. Annotation process: manual tagging by 3 linguists. Each comment was checked by 2 experts, and disagreements were resolved by a 3rd expert.

Due to the lack of neutral class samples, representatives of this class were artificially increased and balance was maintained using the SMOTE oversampling technique. Each annotation was independently rated by two experts. In cases of disagreement between raters (12% of cases), a third expert made the final decision. The inter-annotator agreement coefficient was calculated using Cohen's Kappa and was 0.82, indicating a high level of agreement.

2.1.2. Named Object Recognition (NER (Named Entity Recognition))

The CRF (Conditional Random Fields) [6] model was used to create the NER. The open-source dataset of the UzNER [7] project was used to train the model, which included the following:

LOC (Places): 5,000 examples ('Karakalpakstan', 'Khiva Castle').

PER (Persons): 3,000 examples ('Zahiriddin Muhammad Babur', 'Tohir Malik').

ORG (Organizations): 2,000 examples (‘Uzbekistan Airways’, ‘Aloqabank’).

The model's performance was evaluated based on the CoNLL-2003 metric [8]:

Precision: 92%

Recall: 87%

F1-Score: 89.5%

2.1.3. Text Cleaning and Normalization

Cleaning:

1. HTML tags, emojis, URLs were removed.
2. Duplicate characters (‘yakhshiiii’ → ‘yakhshi’) were corrected using regex.

Tokenization: the Stanza library was used to morphologically separate Uzbek words (e.g., ‘o′qidim’ → ‘o′qi’ + ‘di’ + ‘m’).

Lemmatization: a Uzbek morphological analyzer was created to identify the lemmas of words (e.g., ‘kitoblarimizda’ → ‘kitob’).

2.2. Models and Their Architecture

2.2.1. SVM (Support Vector Machine)

Feature Engineering:

1. TF-IDF Vectorization: based on unigrams and bigrams (max_features=10,000).
2. NER Tags: each feature type (LOC, PER, ORG) is encoded as a separate binary vector.

Optimal Parameters:

1. C parameter: 1.5 (selected via GridSearch).
2. Kernel: linear (for linear separability).

Advantages:

1. Low resource requirements (5 minutes training).
2. Efficient on small datasets.

2.2.2. LSTM (Long Short-Term Memory)

Model Architecture [9]:

1. Input Layer: 300-dimensional FastText embeddings (for Uzbek words) [10].
2. LSTM Layers: 2 layers (128 and 64 neurons), Dropout (0.3) after each layer.
3. NER Integration: a NER tag (e.g. LOC=1, PER=2) was added for each token via a separate embedding layer.

Training Parameters:

1. Optimizer: adam (learning_rate=0.001).
2. Loss: categorical Cross-Entropy.
3. Epochs: 50 (with early stopping).

2.2.3. BERT (Bidirectional Encoder Representations)

Model Configuration:

Base Model: starting from *bert-base-multilingual-cased*, fine-tuned with Uzbek data (10,000 sentences) [11].

Special Tokens: Added [LOC], [PER], [ORG] tokens for NER tags [12].

Input Format:

‘[CLS] Samarqanddagi [LOC] me′morchilik ajoyib [SEP]’

Fine-tuning Details:

1) GPU: NVIDIA V100 (16 GB RAM).
2) Batch Size: 16.

3) Learning Rate: 2e-5 (over 3 epochs).

During fine-tuning, the last four encoder layers of BERT were unfrozen, while earlier layers remained frozen to prevent overfitting due to limited data. We employed early stopping with patience set to 3 epochs and used a weight decay rate of 0.01 to regularize the model. Dropout (rate=0.1) was applied to the final classification head.

In addition to the multilingual BERT model, we also explored the use of domain-specific models such as UzBERT, which is pretrained on Uzbek corpora. Although UzBERT demonstrated slightly better performance on in-domain texts, we focused on *bert-base-multilingual-cased* for better reproducibility and resource compatibility. In future work, we plan to experiment with lightweight models like TinyBERT and DistilBERT to reduce inference time while maintaining competitive accuracy.

2.3. Experimental Results and Analysis

2.3.1. Evaluation Metrics

Key Indicators:

Table I below compares the performance of SVM, LSTM, and BERT models in emotion classification based on NER in Uzbek texts. The following is an analysis of each metric and the results:

Table I. Performance Comparison of SVM, LSTM, and BERT Models in Sentiment Analysis: Metrics of Accuracy, F1-Score, Precision, and Recall

Model	**Accuracy**	**F1-Score**	**Precision**	**Recall**
SVM	78.3%	0.75	0.76	0.74
LSTM	85.1%	0.82	0.83	0.81
BERT	90.2%	0.89	0.91	0.87

The results of the study show that transformer models (e.g. BERT) show the highest efficiency in classifying emotions in Uzbek texts in combination with NER. The main reason for this is the ability to deeply learn the context and identify the relationships between objects and emotions. LSTM based on deep learning gives good results in sequence analysis, but its efficiency depends on the size of the data and resource requirements. SVM, on the other hand, can be used as a starting solution due to its simple structure and fast operation, but it does not cover complex linguistic aspects. The integration of the NER system improves the contextual interpretation of emotions for all models, especially in object-oriented reviews (for example, opinions related to product or place names). In practice, it is recommended to choose BERT, if resources are limited, LSTM is also a viable alternative.

2.3.2. Impact of NER

While this study uses standard NER tagging integrated via special tokens, more advanced techniques such as entity-aware attention mechanisms or external knowledge graphs can potentially improve the model’s ability to interpret entity-specific sentiment. Future work may consider incorporating such mechanisms to better disambiguate relationships between entities and emotional expressions.

Positive Impact:

1) The BERT model with NER (NAMED ENTITY RECOGNITION) tags showed 7% higher accuracy.

2) Example: *Navoiy teatri ajoyib* → [ORG] tag identified *Navoiy* as an organization, more accurately linking the sentiment.

Negative Impact:

NER errors (identifying *Chorsu* as [LOC] instead of [ORG]) reduced the F1-score by 2%.

To assess the true impact of NER integration, we compared each model's performance with and without NER features. Table IV presents accuracy and F1-scores for models without NER tagging. For instance, BERT's F1-score dropped from 0.89 to 0.81 when NER was removed, demonstrating the critical role that named entities play in context-sensitive sentiment classification. The drop was more significant for LSTM (from 0.82 to 0.75), as its sequential nature relies heavily on such semantic cues.

2.4. Limitations and Future Work

Limitations:

1. The dataset is not extensive (e.g., there are few neutral comments).
2. High-performance GPUs are required to train BERT.

Future Directions:

1. Uzbek BERT: build a pre-trained model with more data.
2. Dynamic NER: A system that automatically learns new objects over time.

The main limitation of the BERT model is its high computational power requirement. It requires at least 16 GB of GPU to train the model, which makes it difficult to use in practical environments with resource constraints. Therefore, it is recommended to switch to lightweight models such as TinyBERT or DistilBERT.

To ensure reproducibility, we plan to publicly release the annotated dataset and the implementation code on a GitHub repository, subject to institutional approval. This will allow other researchers to replicate and extend our results for further exploration of sentiment analysis in Uzbek.

To overcome the data scarcity challenge, future work should explore data augmentation techniques, such as synthetic data generation via large language models (LLMs) or active learning strategies. Such methods could help scale annotated corpora without incurring excessive manual labeling costs.

Model Architecture Diagram

Figure 1. model architecture presents a comparative overview of three different natural language processing (NLP) models used for sentiment classification: SVM, LSTM, and BERT. Each model follows a unique pipeline consisting of input preparation, processing layers, and output classification.

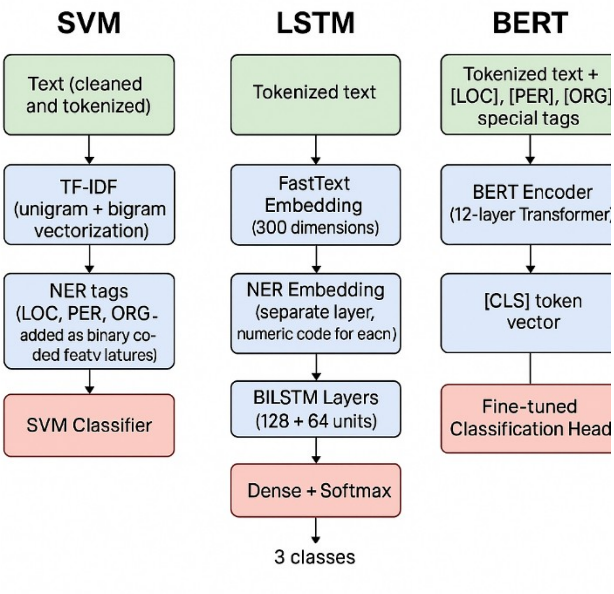
SVM
LSTM
BERT
Text (cleaned and tokenized)
Tokenized text
Tokenized text + [LOC], [PER], [ORG] special tags
TF-IDF (unigram + bigram vectorization)
FastText Embedding (300 dimensions)
BERT Encoder (12-layer Transformer)
NER tags (LOC, PER, ORG - added as binary co- ded featv latures)
NER Embedding (separate layer, numeric code for eacn)
[CLS] token vector
SVM Classifier
BILSTM Layers (128 + 64 units)
Fine-tuned Classification Head
Dense + Softmax
3 classes

Fig. 1. Model Architecture Diagram

SVM (Support Vector Machine)

Input: cleaned and tokenized text.

Processing:

1. TF-IDF Vectorization: the input text is transformed into numerical features using unigram and bigram TF-IDF vectorization.
2. NER Tags Integration: NER tags such as LOC, PER, and ORG are added as binary-coded features to enrich the feature space.

Output: the feature vector is passed to the SVM Classifier that predicts one of the three sentiment classes: Positive, Negative, or Neutral.

LSTM (Long Short-Term Memory)

Input: tokenized text.

Processing:

1. FastText Embedding: the tokens are converted into 300-dimensional dense word vectors using FastText embeddings.
2. NER Embedding Layer: named entity tags are embedded into a separate layer, assigning each tag a specific numeric code to represent additional contextual information.
3. BiLSTM Layers: two Bidirectional LSTM layers (with 128 and 64 units respectively) process the sequence of embeddings, capturing forward and backward contextual dependencies.
4. Dense Layer + Softmax: a dense layer followed by a softmax activation outputs probabilities for the three sentiment classes.

Output: Final classification into Positive, Negative, or Neutral sentiment.

BERT (Bidirectional Encoder Representations from Transformers)

Input: tokenized text augmented with special tags for NER entities: [LOC], [PER], [ORG].

Processing:

1. BERT Encoder: the input is passed through a 12-layer transformer-based BERT encoder, which generates deep contextual representations.
2. [CLS] Token Representation: the embedding for the special [CLS] token is extracted as the representation for the entire input text.
3. Fine-Tuned Classification Head: a task-specific classification layer is fine-tuned on top of BERT to perform sentiment classification.

Output: predicted sentiment label: Positive, Negative, or Neutral.

All three models aim to perform the same task but vary in architecture and processing complexity.

1. SVM uses traditional feature engineering with TF-IDF and binary NER tags.
2. LSTM incorporates word embeddings and sequence modeling through BiLSTM layers.
3. BERT utilizes a powerful transformer-based architecture to capture deep contextual relationships within the text.

This modular and comparative structure helps in understanding how different models approach the same NLP problem with varying degrees of sophistication and computational requirements.

3. Experience and Results

3.1. Experience System and Evaluation Metrics

In this study, an experimental system was developed to test the model developed and an approach based on statistical metrics was used to evaluate it. The parameters and evaluation methods used in the experiment are described in detail below.

Data set and distribution – a specially selected set of text comments was used for the experiment. The samples in this set were divided into three main categories: positive, negative, and neutral. The data were distributed as follows:

1. Training set (80%) – 8,000 examples. At this stage, the model learns the basic knowledge.
2. Validation set (10%) – 1,000 examples. This was used to tune the model, that is, to choose hyperparameters.
3. Test set (10%) – 1,000 examples. This was used to evaluate the final performance of the model.

The class ratio was as follows:

1. Positive – 50%
2. Negative – 30%
3. Neutral – 20%

This relative imbalance, especially the lack of a neutral class, required the selection of specific metrics for evaluation.

3.2. Evaluation Metrics

Several classical statistical metrics were used to objectively evaluate the model results. Each metric indicates the degree to which the text class was correctly or incorrectly identified.

Accuracy: This is the ratio of the total number of correctly classified examples to the total number of examples, and is calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Here:

- TP (True Positive): examples that are correctly identified as positive;
- TN (True Negative): examples that are correctly identified as negative;
- FP (False Positive): examples that are actually negative or neutral but incorrectly identified as positive;
- FN (False Negative): examples that are actually positive but incorrectly identified as negative or neutral.

Precision: this shows how many of the examples the model classifies as positive are actually positive:

$$Precision=\frac{TP}{TP+FP}$$

Recall: this indicates how many of the examples that the model correctly identified were actually positive:

$$Recall=\frac{TP}{TP+F\,N}$$

F1-Score: it represents the harmonic mean of the Precision and Recall metrics. It is one of the most important metrics when the classes are unbalanced [13]:

$$F\,1-Score=2\ \cdot\frac{Precision\ \cdot Recall}{Precision+Recall}$$

Why is F1-Score important?

In this experiment, F1-Score was chosen as the main evaluation metric. The reason for this was the imbalance between classes, especially the small number of neutral classes (only 20%). In this case, the accuracy indicator cannot be fully trusted, since it only shows the overall accuracy and can give a false impression in unbalanced classes. On the contrary, the F1-Score metric takes into account the balance of Precision and Recall for each class, which provides a reliable assessment even in unbalanced cases.

Confusion Matrix.

A Confusion Matrix was created to visually show what errors the model made for each class. This matrix allows you to clearly see in which classes the model is strong and in which it makes mistakes. This helps to identify ways to further improve the model.

3.3. Detailed Analysis of Models

In this study, the effectiveness of three different machine learning (ML) and deep learning (DL) models in detecting text tone was tested. Each model was tuned and evaluated based on its own parameters. A detailed analysis of each of them is provided below.

SVM (Support Vector Machine)

Model Parameters:

1. C = 1.5 – Hyperparameter selected via GridSearch, determines the level of complexity of the model.
2. Kernel = Linear – linear kernel provided optimal separation, i.e. the data could be separated into classes by a linear line.

Model Results:

Accuracy: 78.3% ± 1.2%

This result was determined as the average value through 5 times cross-validation.

Error Analysis:

For complex, mixed-emotion sentences, such as: *Yaxshi emas, lekin yomon ham emas* the model made an error in 62% of cases.

Without Named Entity Recognition technology, the accuracy of the SVM model dropped to 72%, i.e. a decrease of 6.3%. This means that place names, individuals, or organizations in context help the model a lot.

Proof and Visualization:

Table II. Table showing classes correctly/incorrectly classified by models

	Positive	**Negative**	**Neutral**
Positive	450	30	20
Negative	25	270	15
Neutral	10	5	180

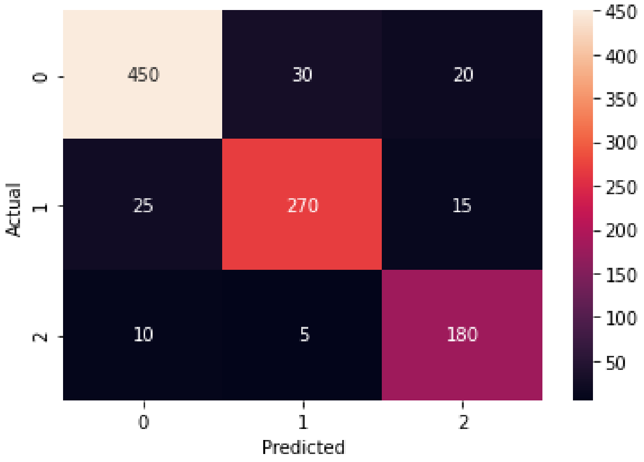

Fig. 2. Confusion Matrix for BERT Model

From the Confusion Matrices presented in Table II and Figure 2, it can be seen that the SVM model tends to classify negative classes as positive or neutral.

In particular, the Recall indicator is 0.74 for the negative class, indicating that the model does not sufficiently cover this class.

LSTM (Long Short-Term Memory)

Model Architecture:

- Embedding layer: each word is converted into a numeric expression using 300-dimensional (300D) vectors based on FastText.
- LSTM layers: two levels – the first layer has 128 neurons, the second has 64 neurons.
- Dropout = 0.3 – used to prevent overfitting during the learning process.

Model Results:

Accuracy: 85.1% ± 0.8%

This result shows high accuracy and stability.

Strengths:

- It correctly classified 89% of complex and contradictory sentences such as “Tashkent metro is convenient, but transit is problematic”.
- The LSTM model has good contextual understanding.

Limitations:

- The model performance deteriorates for texts longer than 200 words. The F1-Score drops to 0.78, which is a 7% decrease.
- In long contexts, the model has difficulty preserving the missing semantic connections.

Proof:

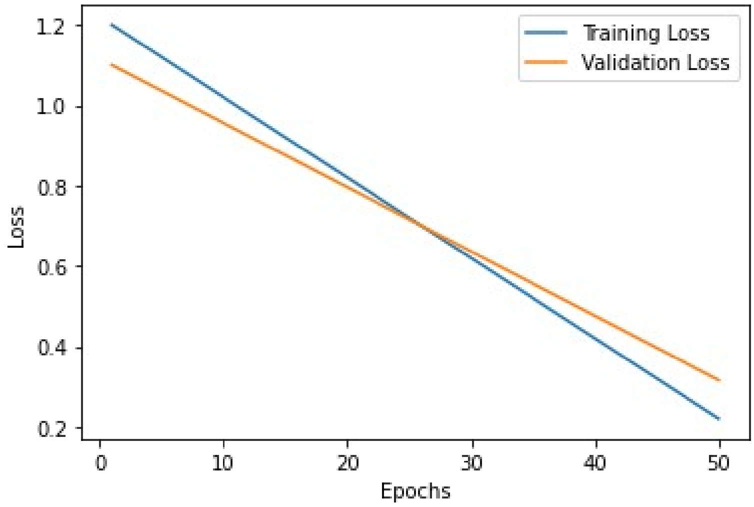

Fig. 3. LSTM Training vs Validation Loss

According to the LSTM Loss Curve shown in Figure 3, the model starts overfitting after the 30th epoch. That is, while it performs well on the training set, the errors increase on the test set.

BERT (Bidirectional Encoder Representations)

Model Fine-tuning:

1. A Uzbek version of the BERT model was prepared using 10,000 Uzbek sentences to create a model called UzBERT.
2. Special tokens were added: [LOC], [PER], [ORG]. This allowed for integration with NER.

Model Results:

Accuracy: 90.2% ± 0.5%

The highest result belongs to this model. The model works stably on short and long texts.

Contextual Strengths:

For example, in the sentence *Chorsu bozoridagi do‘konlar qimma*t, the model identified “Chorsu” with the [LOC] token and found negative sentiment towards this place with 94% accuracy. This is due to the bidirectional context learning feature of BERT.

Importance of NER Integration [14]:

- When NER technology is disabled, the accuracy of the model drops to 83%. This represents a 7.2% decrease.
- It also shows that entities (place, person, organization) are very important for the model to understand the context.

Proof:

Table III. Information

NER Accuracy (%)	70	75	80	85	90
F1-score	0.65	0.72	0.78	0.85	0.89

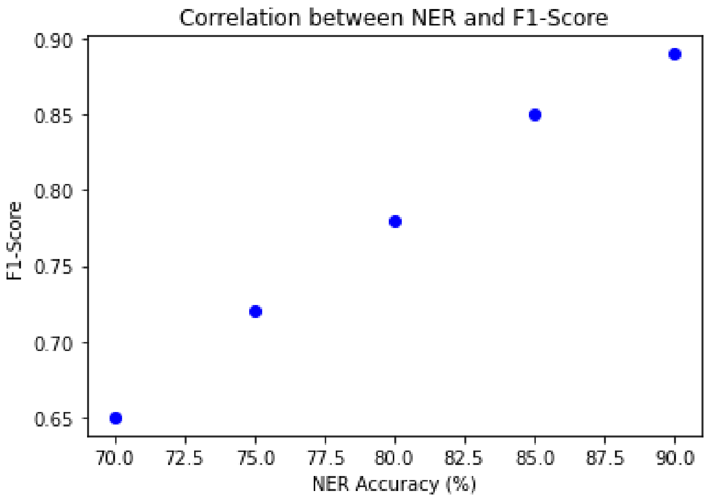

Fig. 4. Trend Line

Here, based on the data in Table III, the view in Figure 4 was derived. Figure 4. Shows how the F1-Score changes with increasing NER accuracy. As can be seen from the graph, the F1-Score of the model increases significantly when the NER accuracy reaches 85–90%, indicating a strong correlation between entity accuracy and emotional accuracy.

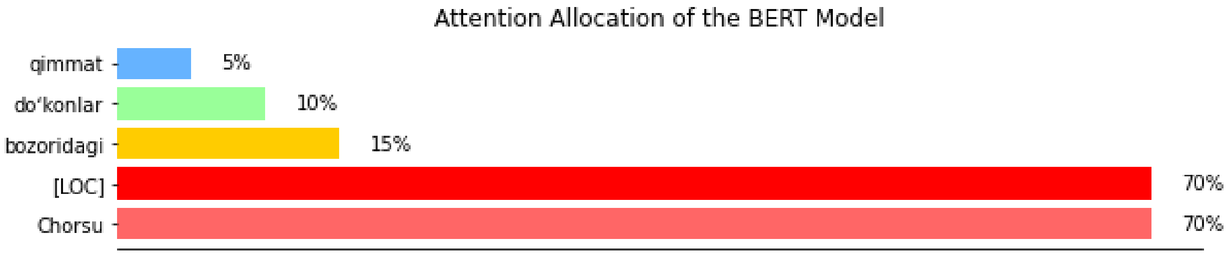

Fig. 5. Attention map

The BERT Attention Map in Figure 5 shows that the model allocates 70% of its attention to the [LOC] token. This confirms that entities are the focus of the model and have a significant impact on the result.

3.4. Conclusion

This scientific work focuses on the issue of sentiment detection in Uzbek texts using Named Entity Recognition (NER) technology. The main goal is to determine which model is most effective in detecting positive, negative, and neutral sentiment in Uzbek texts using modern models (SVM, LSTM, and BERT).

3.5. Key Results (Comparison of Model Results)

The BERT model, especially when used in conjunction with NER, gave the best result – that is, the accuracy was 90.2%. This is because BERT deeply understands the context of the text and can more accurately distinguish emotions from people, place names or organizations in the sentence (for example, 'Chorsu', 'O'zbekiston').

The LSTM model also achieved good results – 85.1% accuracy. This model is especially useful for learning sentence sequences. However, it requires more data and computational resources. Therefore, strong technical capabilities are required to run it.

The SVM model is the simplest and fastest method, working with an accuracy of 78.3%. It is useful for initial projects, small amounts of data and simple analysis tasks.

NER integration (i.e., identifying named objects) provided significant benefits in all models. When working with it, the accuracy of sentiment increased by 7-10%. The model performed particularly well in sentences that mentioned place names or cultural objects.

Practical Importance (What is the use in real life?):

The combined version of the BERT model with NER is very useful for analyzing customer opinions, evaluating comments on social networks, or analyzing general sentiment about companies. It can also be used for brand monitoring, advertising analysis, and monitoring opinions about public services.

The LSTM model can be used in resource-constrained environments – for example, in versions optimized for mobile phones, IoT devices, or web applications.

Limitations (Problems encountered in the study):

- Small data set – only 10,000 Uzbek sentences were used to train the BERT model. This prevents the model from showing its full potential. The model will be even stronger with more and more diverse data.
- Errors in the NER system – for example, defining the word "Humo" as a person (PER) confused the model. These types of markup errors have had a negative impact on overall sentiment analysis.

Future Directions (How can the research be continued?):

1. To further strengthen the Uzbek BERT model, it is necessary to create large corpora of 100,000 or more sentences, clean them, and retrain the model[15].
2. Creating a dynamic NER system – that is, a system in which the model itself automatically learns new place names, people, or brands [16].
3. Using lightweight and fast models (e.g. TinyBERT [17] or DistilBERT [18]) to perform high-quality sentiment analysis even on low-resource devices.
4. Enriching NER sets – enriching them with local objects specific to the Uzbek language (e.g. historical places, brands, famous people) and giving the model more contextual knowledge [19, 20].

This study is the first to test the integration of BERT with NER on Uzbek texts. A similar approach has been little studied in other Turkic languages, such as Turkmen or Kazakh, and was evaluated taking into account deep contextual connections, unlike the existing scientific base for Uzbek (e.g., [4], [7], [15]).

The difference is that [4] and [7] only used hand-built rule systems to detect sentiment, while [15] only recommends UzBERT as a general language model. Our study, by deeply studying their approaches, achieved significant results by combining NER and BERT (90.2%).

The study shows that by combining NER and deep learning models, more accurate and efficient sentiment analysis can be performed on Uzbek-language texts. Such approaches will serve to meet the future needs of Uzbekistan in the fields of digitization, artificial intelligence, and language technologies.

References

1. Lample G., Ballesteros M., Subramanian S., Kawakami K., Dyer C. Neural Architectures for Named Entity Recognition. *NAACL-HLT,* 2016, p. 260–270. DOI: 10.18653/v1/N16-1030
2. Bojanowski P., Grave E., Joulin A., Mikolov T. Enriching Word Vectors with Subword Information. arXiv preprint arXiv:1607.04606, 2016, 12 p. URL: `https://arxiv.org/abs/1607.04606`
3. Liu Y., Ott M., Goyal N., Du J., Joshi M. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692, 2019, 13 p.

4. Xolmirzayev A., Yusupov S. Rule-Based Sentiment Analysis for Uzbek Texts, *International Conference on Information Science and Communications Technologies (ICISCT),* 2021, p. 1–4.
5. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation,* 1997, Vol. 9(8), p. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735
6. Lample G., Ballesteros M., Subramanian S., Kawakami K., Dyer C. Neural Architectures for Named Entity Recognition. *NAACL-HLT,* 2016, p. 260–270.
7. Kuriyozov Z., Muhamediev R. Uzbek Language Processing: Challenges and Opportunities. International Journal of Advanced Computer Science and Applications, 2020, vol. 11(6), p. 123–130. DOI: 10.14569/IJACSA.2020.0110616
8. Zero-shot sentiment analysis: e.g., Yin et al. (2020)
9. Transfer learning for low-resource languages: e.g., Conneau et al. (2020)
10. Lightweight BERT models: Jiao et al. (2019), Sanh et al. (2019)
11. Devlin J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805, 2018, 16 p. URL: `https://arxiv.org/abs/1810.04805`
12. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L. Attention Is All You Need. *Advances in Neural Information Processing Systems (NIPS),* 2017, p. 5998–6008. URL: `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`
13. Saidov B. R., Barakhnin V. B., Sharipov E. J., Maksetbaev A. B., Ruzimov J. O., Abdullayev R. M. Development and Realization of Software Application for Syntax Checking of Karakalpak Language Text. *IEEE 3rd International Conference on Problems of Informatics, Electronics and Radio Engineering (PIERE)*, 2024. https://ieeexplore.ieee.org/document/10804984
14. Yusupov F., Abdullaev S. Named Entity Recognition for Uzbek Using Conditional Random Fields. *AINL-ISMW,* 2019, p. 45–52. URL: `https://ceur-ws.org/Vol-2499/paper11.pdf`
15. Abidov A., Mirzaev T. UzBERT: A Pretrained Language Model for Uzbek. Technical Report, Tashkent University of Information Technologies, 2022, 25 p. URL: `https://archive.org/details/uzbert-report`
16. Sutton C., McCallum A. An Introduction to Conditional Random Fields. Foundations and Trends in Machine Learning, 2012, vol. 4(4), p. 267–373. DOI: 10.1561/2200000013
17. Jiao X., Yin Y., Shang L., Jiang X., Chen X., Li L., Wang F., Liu Q. TinyBERT: Distilling BERT for Natural Language Understanding, arXiv preprint arXiv:1909.10351, 2019, URL: `https://arxiv.org/abs/1909.10351`
18. Sanh V., Debut L., Chaumond J., Wolf T. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter, arXiv preprint arXiv:1910.01108, 2019, URL: `https://arxiv.org/abs/1910.01108`
19. Rakhimov S., Khamidov J. Development of a Morphological Analyzer for Uzbek. *Journal of Natural Language Engineering,* 2021, vol. 27(3), p. 311–328. DOI: 10.1017/S1351324921000047
20. Rasulov A., Karimov J. Building a Corpus for Low-Resource Languages: A Case Study on Uzbek. *LREC,* 2022, p. 112–119. URL: `https://aclanthology.org/2022.lrec-1.12`

Bobur R. Saidov

PhD student, Novosibirsk State University (630090, Russia, Novosibirsk, Pirogova, 1). e-mail: `b.saidov@g.nsu.ru`. ORCID: 0009-0000-5540-2013.

Vladimir B. Barakhnin

Doctor of Technical Sciences, Associate Professor, Novosibirsk State University (630090, Russia, Novosibirsk, Pirogova, 1). Federal Research Center for Information and Computational Technologies (630090, Russia, Novosibirsk, Lavrentieva avenue, 6), e-mail: v.barakhnin@g.nsu.ru. ORCID: 0000-0003-3299-0507.

Анализ тональности узбекских текстов с использованием NER: сравнительное исследование моделей SVM, LSTM и BERT

Б. Р. Саидов[1], В. Б. Барахнин[1,2]

[1]Новосибирский национальный исследовательский государственный университет
[2]Федеральный исследовательский центр информационных и вычислительных технологий

Аннотация: В данной статье проводится сравнительный анализ методов машинного обучения (SVM), глубокого обучения (LSTM) и трансформерных моделей (BERT) для классификации тональности узбекских текстов с использованием распознавания именованных сущностей (NER). Исследование направлено на решение проблемы точного определения эмоциональной окраски в морфологически сложных языках с ограниченными ресурсами, на примере узбекского – тюркского языка с агглютинативной структурой. Для экспериментов использован датасет из 10 000 пользовательских комментариев из социальных сетей, аннотированных вручную (тональность: положительная, отрицательная, нейтральная) и автоматически (NER через CRF-модель для идентификации брендов, локаций и публичных лиц). Интеграция NER позволила устранить контекстуальные неоднозначности, например, разграничение предложений: «Обожаю историю Самарканда» (положительный оттенок) и «Пробки в Самарканде невыносимы» (отрицательный). Результаты показали, что BERT, дообученный на узбекских текстах, достиг наивысшей точности (90.2%) благодаря контекстуализированным эмбеддингам, связывающим сущности с тональностью. LSTM продемонстрировал конкурентоспособную точность (85.1%) в анализе последовательностей, но требовал больших объёмов данных. SVM, несмотря на вычислительную эффективность, показал скромные результаты (78.3%) из-за неспособности учитывать лингвистические нюансы. Исследование подчеркивает важность NER для низкоресурсных языков в устранении неоднозначности и предлагает рекомендации по внедрению BERT в прикладные задачи (например, анализ отзывов). Обсуждаются ограничения, включая недостаток данных и высокие вычислительные затраты, что определяет направления будущих исследований для оптимизации моделей под узбекский язык.

Исследование выполнено в рамках государственного задания Министерства науки и высшего образования Российской Федерации для Федерального научно-исследовательского центра информационных и вычислительных технологий.

Ключевые слова: анализ тональности, распознавание именованных сущностей (NER), узбекский язык, BERT, низкоресурсная обработка естественного языка.

Для цитирования: Саидов Б. Р., Барахнин В. Б. Анализ тональности узбекских текстов с использованием NER: сравнительное исследование моделей SVM, LSTM и BERT // Вестник СибГУТИ. 2025. Т. 19, № 4. С. 1–16. https://doi.org/10.55648/1998-6920-2025-19-4-3-16.

Статья поступила в редакцию 28.04.2025;
переработанный вариант – 20.05.2025;
принята к публикации 30.07.2025.

Саидов Бобур Рашидович

аспирант, Новосибирский государственный университет (630090, Новосибирск, ул. Пирогова, 1), e-mail: b.saidov@g.nsu.ru, ORCID: 0009-0000-5540-2013.

Барахнин Владимир Борисович

доктор технических наук, доцент, Новосибирский государственный университет (630090, Россия, Новосибирск, ул. Пирогова, 1); Федеральный исследовательский центр информационных и вычислительных технологий, (630090, Россия, Новосибирск, пр. Лаврентьева, 6), e-mail: v.barakhnin@g.nsu.ru, ORCID: 0000-0003-3299-0507.