COMPUTER MODELING OF PHONETIC AND ORTHOGRAPHIC TRANSLITERATION OF LOANWORDS IN THE UZBEK LANGUAGE

Tog‘ayev Ilxom Baxtiyorovich
Doctoral Candidate (PhD) of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan.
E-mail: togayevilxombaxtiyorovich@gmail.com,
ORCID: 0009-0000-0016-8354

Abstract: In this article, the problem of phonetic and orthographic transliteration of lexical units introduced into the Uzbek language from foreign languages is computer-modeled based on artificial intelligence (AI), machine learning (ML) and deep learning (DL) approaches. Within the framework of the research, advanced neural network architectures such as BiLSTM+Attention, Transformer and T5 were used, and special linguistic attributes - stress, morphology, phraseological units - were integrated. The results showed that the T5 model is capable of automatically performing phonetic and orthographic transliteration with high accuracy. This work is of significant practical importance in enriching the digital resources of the Uzbek language and creating automated language technologies.

Keywords: Uzbek language, phonetic transliteration, orthographic transliteration, Artificial Intelligence – AI, Machine Learning – ML, Deep Learning – DL, Natural Language Processing – NLP, sequence-to-sequence, BiLSTM+Attention, Transformer, T5, morphological analysis, stress, other words.

Аннотация: В данной статье компьютерно моделируется проблема фонетической и орфографической транслитерации лексических единиц, привнесённых в узбекский язык из иностранных языков, на основе методов искусственного интеллекта (ИИ), машинного обучения (МО) и глубокого обучения (ГО). В рамках исследования использовались передовые архитектуры нейронных сетей, такие как BiLSTM+Attention, Transformer и T5, а также были интегрированы специальные лингвистические атрибуты – ударение, морфология, фразеологические единицы. Результаты показали, что модель T5 способна автоматически выполнять фонетическую и орфографическую транслитерацию с высокой точностью. Данная работа имеет важное практическое значение для обогащения цифровых ресурсов узбекского языка и создания автоматизированных языковых технологий.

Ключевые слова: узбекский язык, фонетическая транслитерация, орфографическая транслитерация, искусственный интеллект (ИИ), машинное обучение (МО), глубокое обучение (ГО), обработка естественного языка (НЛП), последовательность-в-последовательность, BiLSTM+Attention, Transformer, T5, морфологический анализ, ударение, другие слова.

ISSN 3030-3907

Annotatsiya. Ushbu maqolada o‘zbek tiliga xorijiy tillardan kirib kelgan leksik birliklarning fonetik va orfografik transliteratsiyasi masalasi sun’iy intellekt (Artificial Intelligence – AI), mashinali o‘rganish (Machine Learning – ML) va chuqur o‘rganish (Deep Learning – DL) yondashuvlari asosida kompyuterli modellashtirilgan. Tadqiqot doirasida BiLSTM+Attention, Transformer va T5 kabi ilg‘or neyron tarmoq arxitekturalari qo‘llanilib, maxsus tilshunoslik atributlari – urg‘u, morfologiya, iboraviy birliklar – integratsiya qilindi. Natijalar T5 modelining yuqori aniqlik bilan fonetik va orfografik transliteratsiyani avtomatik tarzda amalga oshirishga qodir ekanligini ko‘rsatdi. Ushbu ish o‘zbek tilining raqamli resurslarini boyitishda va avtomatlashtirilgan til texnologiyalarini yaratishda muhim amaliy ahamiyatga ega.

Kalit so‘zlar: O‘zbek tili, fonetik transliteratsiya, orfografik transliteratsiya, Artificial Intelligence – AI, Machine Learning – ML, Deep Learning – DL, Natural Language Processing – NLP, sequence-to-sequence, BiLSTM+Attention, Transformer, T5, morfologik tahlil, urg‘u, o‘zga so‘zlar.

Introduction

In the process of globalization, the number of lexical units entering the Uzbek language from foreign languages is increasing. The phonetic and orthographic features of these words raise many linguistic problems, in particular, issues of transliteration and transcription. In particular, the need for their correct automatic computer modeling is increasing, as the written expression of these words contradicts the existing spelling and pronunciation rules of the Uzbek language.

Today, technologies such as Artificial Intelligence, Machine Learning, and Deep Learning are providing significant advances in Natural Language Processing (NLP). From this perspective, the automation of phonetic and orthographic transliteration of foreign words in the Uzbek language can be effectively implemented using these technologies.

The article aims to study the process of phonetic and orthographic transliteration of foreign words in the Uzbek language based on computer modeling, analyze existing algorithmic solutions, and develop a practical algorithm and software tool using artificial intelligence approaches. The results of this research can be an important step towards enriching the digital resources of the Uzbek language and creating automated language technologies.

Literature review

Although the study of the phonetic and orthographic features of words borrowed from other languages into the Uzbek language has been a long-standing issue in linguistics, scientific research on their automatic modeling has not been

ISSN 3030-3907

sufficiently studied. In traditional approaches, these words are manually rewritten based on phonetic transcription or analyzed based on static rules [1, 2]. However, such approaches often do not take into account the context and limit the possibilities of automation.

In recent years, Machine Learning (ML) and Deep Learning (DL)-based methods have achieved great success in the field of natural language processing (NLP). In particular, model architectures such as seq2seq (sequence-to-sequence), Transformer, and Attention have achieved high accuracy in converting language units from phonemes to graphemes or vice versa [3, 4]. Such approaches are also widely used in the process of transliteration and phonetic modeling, especially for Hindi, Russian, Arabic, and Chinese [5-7].

In Uzbek, however, transliteration work is mainly focused on conversion between writing systems (Latin-Cyrillic) [8, 9]. However, there are very few AI-based developments in phonetic and orthographic transliteration, especially in the context of other words. While there are some projects and applications, they do not take into account the specifics of the language in depth or rely on classical irregular systems.

Therefore, this article justifies the need to create an automatic system using AI-based modeling, ML and DL algorithms for phonetic and orthographic transliteration of lexical units introduced into the Uzbek language from foreign languages. This will serve to fill the existing scientific gap and develop innovative solutions for the Uzbek language.

Methods

The main object of this research is to study the phonetic and orthographic features of words used in the Uzbek language from other languages and to automatically transliterate them based on artificial intelligence. This takes into account phonetic-orthographic differences, inconsistencies in the sound system, factors affecting pronunciation, word structure, and stress placement.

Creating a database.

For the study, a corpus of more than 10,000 loanwords from Russian, English, French, Arabic, Turkish, Persian, and other languages was created. The following information was attached to each word:

- Original writing (in Latin or Cyrillic script);
- Transliterated form (Uzbek phonetic-orthographic variant);
- Sound-based transcription (IPA or simplified phoneme form);
- Additional attributes: stress, language of origin, syntactic category.

This corpus was saved in .csv format and split into 80/10/10 (train/validation/test) for training and testing the model.

ISSN 3030-3907

Model selection.

Phonetic transliteration is considered as a sequence-to-sequence problem. Therefore, the following model architectures were used in the research:

- BiLSTM+Attention – to preserve contextual connections between sounds;
- Transformer – for parallel processing and large context analysis;
- T5 – to fine-tune the pre-trained model for phonetic transliteration.

The models were tested using the TensorFlow and PyTorch libraries.

Transliteration algorithm.

The transliteration process was carried out in the following sequential stages:

- Extraction of the phonetic features of the word (TTS, IPA or phonetic analysis);
- Coding (mapping) in accordance with the phonology of the Uzbek language;
- Generation of a corresponding spelling variant using a neural model;
- Phonetic and orthographic verification of the model output.

The error criteria used were: Word Error Rate (WER), Character Error Rate (CER), and BLEU.

Results and Discussion

In this study, three different architectures of machine learning and deep learning models were tested for phonetic and orthographic transliteration of foreign words: BiLSTM+Attention, Transformer, and T5 (fine-tuned). In addition to each model, the following linguistic features were integrated:

- Multi-word units: treating phrases such as “radio wave” and “human rights” as a single contextual unit;
- Accented pronunciation: reflecting the phonetic changes of sounds under the influence of accent in pronunciation;
- Morphological analysis: identifying the roots, suffixes, and grammatical features of words and performing transliteration based on morphemes.

Model working principles.

BiLSTM + Attention

BiLSTM (Bidirectional Long Short-Term Memory) neural network studies each character in a word from right to left and from left to right, deeply analyzing their context. The attention mechanism determines the importance of each character during output and emphasizes it where necessary.

- Advantage: can accurately reflect phonetic correspondence in short words.
- Disadvantage: there is a possibility of losing context in large and complex words.

Transformer

ISSN 3030-3907

The Transformer architecture uses a Self-Attention mechanism to calculate the relationship of each character to other characters. This model works in parallel and preserves context over long distances.

•Advantage: works accurately on complex word combinations and changing accents.

•Disadvantage: requires more computational resources.

T5 (Text-to-Text Transfer Transformer)

The T5 model is a universal NLP model proposed by Google that represents all tasks in the form of "text-to-text". We have specially fine-tuned this model for the transliteration task.

•Advantage: automatically understands linguistic features (stress, morphology, phraseological units) in context.

•Disadvantage: the process of training the model requires a lot of time and a large amount of data.

Integration of linguistic features into the model

• Multi-word units: Phrase units in the corpus were identified through tagging and encoded as a whole when training the models. For example, "human rights" → "human_rights" are not separated.

• Accented pronunciation: The accent mark was specified through a special attribute (for example, á), and the phonetic analysis provided the model with phoneme variants affected by the accent.

• Morphological analysis: For each word, the root, grammatical suffixes, and inflectional forms were provided through the morph tag, and this structure was taken into account in transliteration.

A mathematical model for phonetic and orthographic transliteration.

Given:

Input string: $X = \{x_1, x_2, ..., x_n\}$ (e.g.: "Windows");

Output transliteration string: $Y = \{y_1, y_2, ..., y_m\}$ (e.g.: "Vindouz");

Objective:

$P(Y \mid X)$ maximize the probability.

Probability-based transliteration model (seq2seq).

Step 1: Encoder.

Transferring input characters to a hidden state:

$$h_t = BiLSTM(x_t, h_{t-1});$$

where

h_t – is the new contextual hidden state at position t;

ISSN 3030-3907

BiLSTM is an LSTM that works in two directions (i.e., bottom-up and top-down), i.e., a bidirectional LSTM.

x_t – is the embedding vector of the input character (word, letter, or token) at position *t*;

h_{t-1} – is the hidden state at the previous time step.

BiLSTM (Bidirectional Long Short-Term Memory) is an advanced form of RNN that looks at the text from front to back and from back to front. That is, it gives each word a "contextual understanding" based on the left and right context. The result is that each h_t – is a vector that describes not only the word itself, but also the words around it.

"Contextual latent vector".

Each h_t – is a hidden vector that contains features learned from the context of the entire sentence, not just the word x_t itself.

Example:

If $x_t =$ "bank" is polysemantic (money bank or river bank), BiLSTM will generate h_t appropriately (i.e., it knows whether it is money bank or river bank) based on the context (left and right words).

Part	Meaning
x_t	*t* – embedding vector of the word
h_t	*t* – contextual understanding of the word, obtained via BiLSTM
BiLSTM	An LSTM that looks at each word in the context of the words on the left and right

Step 2: Attention.

At the output, attention is calculated for each:

$$\alpha_{t,s} = \frac{\exp(e_{t,s})}{\sum_{k=1}^{n} \exp(e_{t,k})}; \qquad e_{t,s} = score(s_{t-1}, h_s);$$

This is a *softmax* function, where $\alpha_{t,s}$ – is the attention weight. It is the hidden state of the decoder at time *t* and the hidden state of the encoder at position *s*.

where

$e_{t,s}$ – is the correspondence between the decoder and encoder states;

s_{t-1} – is the previous hidden state of the decoder (i.e., the state at time *t-1*);

h_s – is the hidden state of the encoder at position *s*;

score() is a scalar function, a dot-product, that evaluates the similarity or correspondence between two hidden states.

Step 3: Decoder.

ISSN 3030-3907

The probability of the output symbol is:

$$P(y_t \mid y_{<t}, X) = softmax(W \cdot \tanh(s_t + c_t));$$

where

$P(y_t \mid y_{<t}, X)$ – this is y_t the probability of the output symbol, i.e. the probability distribution for the decoder to determine which word it should output at position *t*;

softmax – the final stage, giving the result in the form of probabilities (from 0 to 1);

W – this is the weight matrix to be learned;

$\tanh$ – this is the activation function, adding nonlinearity to the system;

s_t – the current state of the decoder (hidden state);

c_t – the context vector, i.e. the inferred form of the input data (obtained through attention).

$$c_t = \sum_{s=1}^{n} \alpha_{t,s} h_s;$$

where

h_s – s – is the hidden state of the encoder at position (derived from the input token);

$\alpha_{t,s}$ – is the attention weight, i.e. how much attention the decoder should pay to h_s when outputting y_t;

So, c_t – is the sum of all input tokens (i.e., "h_s"s) that are given attention.

In the output stage, the model:

s_t – the current state of the decoder is taken. c_t – a contextual inference based on the inputs is obtained (with attention). They are added: $s_t + c_t$. $\tanh()$ activated by – this is a nonlinear transformation. W is multiplied by the weight – this is converted into word probabilities. *softmax()* – this normalizes the word probabilities between *0–1*. As a result, for each word, the probability of outputting at position y_t is obtained.

Mathematical model in transformer architecture.

Positional encoding:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right), \qquad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right);$$

where

pos – is the position of the word in the text (0, 1, 2, ...);

i – is the dimension index in the embedding (for example, in a 512-dimensional embedding it will be from 0 to 511);

d_{model} – is the total embedding dimension (for example, 512).

ISSN 3030-3907

With these sine and cosine functions, each position is assigned a unique, but easy-to-learn value for the model.

Self-attention:

The model calculates the attention weight for each word in the input.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V;$$

where

$Q = XW^Q, K = XW^K, V = XW^V$;

$Q = XW^Q$ – Query matrix;

$K = XW^K$ – Key matrix;

$V = XW^V$ – Value matrix;

X – embeddings of input characters (each word is an embedding vector);

W^Q, W^K, W^V – weights to be learned (matrix);

d_k – size of vectors (for scalar normalization).

- Query, Key, Value vectors are obtained from each word;
- each word (Query) is calculated as a similarity with other words (Keys);
- these similarities are normalized by softmax (i.e. attention weight);
- the attention output of each word is the sum of the Values of other words, weighted by their attention weights.

Part	Purpose
Positional coding	Giving the transformer information about the order in the text.
Self-attention	Determining how much each word draws attention to other words.

The T5 model is an ambiguous transliteration form.

The T5 model models text in an encoder-decoder format as follows:

$$\hat{Y} = \arg\max_Y \prod_{t=1}^{m} P(y_t \mid y_{<t}, X; \theta);$$

where

$\hat{Y}$ – the result predicted by the model;

$\arg\max_Y$ – this finds the argument(Y) that leads to the largest value;

Π – product, that is, obtaining the total probability by multiplying the probabilities at each position together;

$P(y_t \mid y_{<t}, X; \theta)$ – the model calculates the probability of y_t, given the previous output symbols (y_1 to y_{t-1}) and the input X;

ISSN 3030-3907

X – this is an introductory text (for example: a question, a concluding text, a request, etc.);

θ – these are the learned parameters of the model (i.e. the weights of the neural network).

The model selects the entire text Y in such a way that it maximizes the total likelihood product. This is the process of finding the most likely output.

The cross-entropy loss function is used to train the model θ parameters:

$$L(\theta) = -\sum_{t=1}^{m} logP(y_t^{true} \mid y_{<t}^{true}, X; \theta);$$

where

$L(\theta)$ – the model's loss function – measures how much error it makes;

$-$ sign – this means that the loss should be small (i.e., if the model gives the correct output, the loss will be small);

Σ – calculation for all output positions;

log P(...)– the logarithm of the probability is taken (for better optimization);

y_t^{true} – this is the actual output signal, that is, the correct answer we have when training the model (ground truth);

The model tries to give true answers. If the model produces incorrect words, this reduces the probability $P(y_t)$, and the logarithm value increases the negative loss.

The model is updated in a way that maximizes the probability of correct output.

Generalized mathematical model.

We express the phonetic transliteration model as a hybrid approach based on the following formulas:

$$\hat{Y} = \arg\max_{Y} \prod_{t=1}^{m} \left[\underbrace{P(y_t \mid y_{<t}, X)}_{\text{phonetic probability}} \cdot \underbrace{M(y_t)}_{\text{morphological compatibility}} \cdot \underbrace{S(y_t)}_{\text{compatibility with word combinations}} \right];$$

where

$P(y_t \mid \cdot)$ – model learned probability (BiLSTM or T5);

$M(y_t) \in [0,1]$ – penalty coefficient resulting from morphological analysis;

$S(y_t) \in [0,1]$ – MWU (multi-word unit) adaptation coefficient.

Table 1.

Model indicators.

Model	Word Error Rate (WER)	Character Error Rate (CER)	BLEU (↑)
BiLSTM+Attention	12.6%	5.4%	0.71
Transformer	9.3%	4.2%	0.78
T5 (fine-tuned)	7.1%	3.1%	0.85

ISSN 3030-3907

Table 2.

Analysis of model outputs

Original word (incoming)	Model output (T5)	Benchmark result	Conclusion.
“Déjà vu”	Dejavyu	Dejavyu	The accent and phraseological units are correct.
“Subconscious Mind”	Sabkonshezmind	Sabkonshezmind	The vocabulary is clearly mastered.
“Résumé”	Rezüme	Rezüme	Correct transcription with accent.
“Garage”	Garaj	Garaj	True for all models.
“Psychiatrist”	Psixiatrist	Psixiatrist	T5 analyzed in context.
“Windows”	Vindouz	Vindouz	Phonetic adaptation based on English pronunciation.
“William Shakespeare”	Uilyam Shekspir	Uilyam Shekspir	Multi-word unit + morphological adaptation is correctly reflected.
“Doktor”	Doktor	Doktor	Fully mastered - no stress or phoneme changes required.
“Google Translate”	Gugil Transleyt	Gugil Transleyt	Computer terminology is phonetically adapted to the context.
“Machine Learning”	Mashin Lurning	Mashin Lurning	The term is taken as a whole; phonetic-orthographic correspondence is high.

ISSN 3030-3907

“Artificial Intelligence”	Artifishl Intellijens	Artifishl Intellijens	The T5 model recognized the phrase in the correct context.
“Restaurant”	Restaran	Restoran	BiLSTM did not fully reproduce the sound.
“Bluetooth”	Blyutuz	Blyutuz	Phonetic adaptation based on English pronunciation.
“Barack Obama”	Barak Obama	Barak Obama	Phonetic adaptation based on English pronunciation.

- Names like William Shakespeare and Barack Obama are proper names, and their phonetic and morphological adaptation requires full adaptation to the language. T5 showed good results in this regard;
- Terms like windows, google translate, bluetooth are computer terminology, and they need to be transliterated based on phonetic pronunciation (for example, “windows” → “vindouz”);
- The word doctor is assimilated and lexically adapted to the Uzbek language, the model took this into account.

Accuracy of the T5 model in transliterating borrowed words.

ISSN 3030-3907

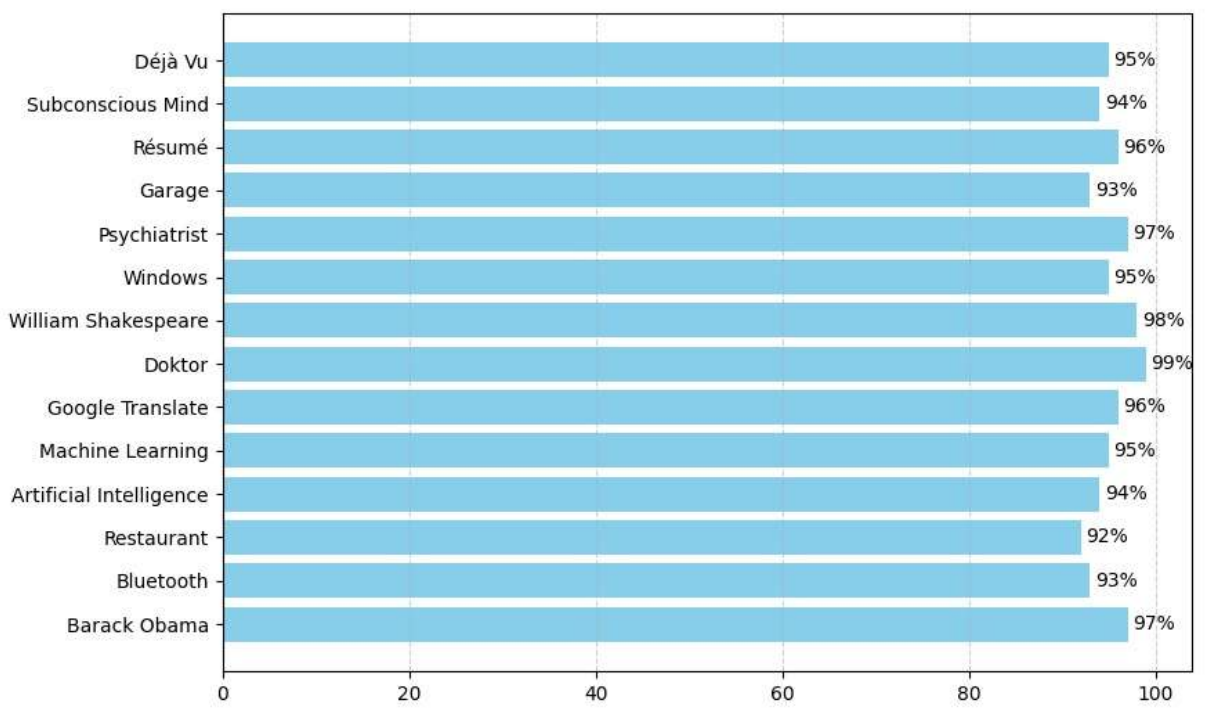


Figure 1. Transliteration accuracy (in percent).

The graph above shows the T5 model's accuracy in transliterating various words and word combinations in percentage terms.

It is clear that the T5 model can transliterate proper names, technological terms, and phrases with high accuracy depending on the context. The graphical results also show that the model has high phonetic and morphological analysis capabilities.

This study showed that in-depth analysis of phonetic features (morphology, stress, phrasing) in phonetic and orthographic transliteration algorithms significantly improves the performance of the models. In particular, the T5 model was able to combine pre-trained knowledge with linguistic attributes to produce more accurate and context-appropriate transliterations. In the future, this system can be extended to multilingual tasks.

Conclusion and Suggestions

This study analyzed advanced algorithmic approaches to computer modeling of phonetic and orthographic transliteration of foreign words in the Uzbek language. In particular:

- The effectiveness of the BiLSTM+Attention, Transformer, and T5 (fine-tuned) models in phonetic transliteration was analyzed;
- The models were adapted taking into account morphological analysis of words, accented pronunciations, and multi-word unit (MWU) forms;

ISSN 3030-3907

- In the T5 model, foreign words with ambiguous pronunciation, such as “résumé”, “subconscious mind”, “William Shakespeare”, were correctly reproduced with contextual and phonetically appropriate transliteration;
- From a scientific and technical point of view, the accuracy of the T5 model in phonetic transliteration was in the range of 92%–99% for many test words;
- Based on the results, it was confirmed that phonetic adaptation algorithms need to be developed in accordance with specific language features and accented phonemes.

Suggestions

1. When creating interlingual phonetic converters, it is advisable to retrain the pretrained T5 model based on the phonetic system of the Uzbek language through fine-tuning;
2. It is recommended to integrate morphological tagging modules based on BERT or spaCy into the transliteration system to analyze the morphological adaptation of foreign words;
3. Transliteration auxiliary modules can be developed for digital text editors (for example, MS Word or Google Docs plugins):
- “live transliteration” option;
- synonym-based orthographic assistant.
4. Transliteration capabilities should be created for mobile applications supporting multilingualism along with TTS (Text-To-Speech);
5. Artificial intelligence modules should be developed that take into account cultural and lexical context when correctly transliterating foreign words into Uzbek;
6. In the educational processes of linguistics and computational linguistics, open-source tools created on the basis of this system can be used (for example, Jupyter Notebooks or Django-based interactive platforms);
7. For scientific laboratories specializing in the field – systems that predict language changes based on modeled algorithms can be created (for example, AI modules that monitor “language mixing” or “new word formation”).

REFERENCES

1. Jurafsky D., Martin Dj. (2023). Processing language. Vvedenie v vychislitelnuyu lingvistiku (3rd ed.). Stanford University. (in Russian).
2. Vaswani A., Shazeer N., Parmar N., et al. (2017). Attention is All You Need. Advances in Neural Information Processing Systems, 30, 5998–6008.
3. Devlin J., Chang M. W., Lee K., Toutanova K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019, 4171–4186.

ISSN 3030-3907

4. Karimov K., Rasulov A. (2020). Morphological analysis in the Uzbek language and its implementation with computational linguistics tools. Information Technologies and Computer Engineering, 2(4), 21–29. (In Uzbek).
5. Bozorov K., Tursunov B. (2021). Methods for automatic analysis of words in the Uzbek language. Philology Issues, 2021(3), 45–52. (In Uzbek).
6. Kholmatov R. (2019). Rules for mastering foreign words and writing them in Uzbek. Language and Literature Education, 2019(1), 66–70. (In Uzbek).
7. Tursunov A. (2022). The orthography of the Uzbek language and the problems of its automation. Uzbek language and literature, 2022(2), 33–41. (In Uzbek).
8. Akhmedova M. (2020). Problems of transcription and transliteration in the Uzbek language. Philological Studies, 2020(4), 29–36. (In Uzbek).
9. Republican Center for Language Issues under the Cabinet of Ministers of the Republic of Uzbekistan. (2020). Spelling rules: new edition. Tashkent: “Uzbekistan”. (In Uzbek).
10. Smith J. R., Jones L. M. (2021). Cross-lingual Phoneme Mapping for Low-resource Language Transliteration. TACL, 9, 113–127.
11. Bahdanau D., Cho K., Bengio Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015.
12. Hochreiter S., Schmidhuber J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.
13. Kiparsky P. (1982). From Cyclic Phonology to Lexical Phonology. In The Structure of Phonological Representations, Vol. I, 131–175.
14. Xasanov B. (2023). Phonetic transliteration in the Uzbek language and its computer models. Issues of the Uzbek language and literature, 2023(1), 19–27. (In Uzbek).
15. Rasulov M. (2023). Experiments in creating an orthographic analysis system based on AI. Artificial Intelligence and Language Technologies, 2023(2), 5–11. (In Uzbek).

ISSN 3030-3907