Innovative: International Multi-disciplinary Journal of Applied Technology
(ISSN 2995-486X) VOLUME 03 ISSUE 5, 2025

THE PROBLEM OF COMPUTER MODELING OF ORTHOGRAPHIC TRANSLITERATION

Abdisait M. Norov*[1], Ilxom B. Tog'ayev [2]

[1,2] Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

*E-mail: abdusaidnorov@gmail.com

Abstract:

Orthographic transliteration is the process of transliterating text based on the spelling rules of a specific natural language. The article examines the problems of transliterating text in the Cyrillic-Latin and Latin-Cyrillic directions and its computer modeling based on the rules of Uzbek spelling.

Keywords: Transliteration, Orthographic Transliteration, Latin, Cyrillic, F1-score, BLEU (Bilingual Evaluation Understudy), CER (Character Error Rate) va WER (Word Error Rate)

1. Introduction

As a result of the continuous improvement of applied linguistics and computational linguistics methods, the need for automatic processing of writing systems in different languages is significantly increasing. Today, within many Turkic languages, the Cyrillic and Latin alphabets are used in parallel. It is known from experience that spelling errors occur when transliterating between these two alphabets. This situation naturally requires a two-stage editing process. From this point of view, computer modeling of transliteration based on spelling rules, in contrast to traditional transliteration, is a pressing issue, and during our research we found it appropriate to call this transliteration "orthographic transliteration".

It is known that the spelling rules of the Uzbek language in the Latin script were adopted on the basis of phonetic, morphological and formal changes, which are different from those in the Cyrillic script. Therefore, it is natural that spelling (orthographic) errors will occur in practice during the transliteration process. This ultimately creates a serious problem in the automatic processing of the Uzbek language or in the creation of artificial intelligence models for the Uzbek language.

From this point of view, the main goal of this research is to develop computer models for

automating orthographic transliteration and improving its quality.

Literature Review

Many studies have been conducted on the issue of transliteration, proposing various approaches.

In particular, K. Knight and J. Graehl (1997) propose a unique method of replacing the word or text being transliterated with its phonetic equivalents and at the same time performing automatic transliteration on a machine. Indeed, it is quite difficult to translate names and technical terms into languages with different alphabets and sound elements. These elements are usually transliterated by replacing them with approximate phonetic equivalents. For example, the English word “computer” is pronounced “コンピューター” in Japanese, that is, “konpyūtā”. Translating such words from Japanese into English is even more difficult, since the transliterated words constitute the bulk of the textual expressions that are not found in bilingual dictionaries [**Error! Reference source not found.**].

A transliteration method based on the methods developed by Andrew Finch and Eiichiro Sumitalar (2008) for direct phrase-based statistical machine translation is proposed. The main goal of this work is to provide a transliteration system that can be used to translate unknown words in a speech-to-speech machine translation system. According to the proposed method, the system should be able to transliterate any sequence of characters, not just words that are already in a pre-defined dictionary [**Error! Reference source not found.**].

B. Mansurov and A. Mansurov, proposed a controlled data-driven approach to transliterating Uzbek dictionary words from Cyrillic to Latin and vice versa as a new method for creating machine transliteration texts for the Uzbek language with limited resources [**Error! Reference source not found.**].

U. Salaev, E. Kuriyozov, and Gómez-Rodríguez Carlos, present a machine transliteration tool between three common scripts used in the low-resource Uzbek language: old Cyrillic, currently official Latin, and the newly announced new Latin alphabets [**Error! Reference source not found.**].

The book "Spelling Dictionary of the Uzbek Language in the Cyrillic and Latin Alphabets" by T. Togayev, G. Tavaldiyeva and M. Akromova, shows that the spelling rules of the Uzbek language in the Latin script have unique phonetic, morphological and formal characteristics compared to the Cyrillic script [**Error! Reference source not found.**].

2. Methods

The Although a number of studies have been conducted on transliteration for Uzbek words, it can be seen that the main difficulties encountered in this work are analyzed in terms of spelling errors. However, the existing methods are based on a small number of rules or manually defined mapping algorithms, and their level of automation is not high enough.

The uniqueness of this study is that it offers not only literal transliteration, but also a model that allows for automatic correction of spelling errors. In this:

a. An approach that allows for automatic detection and correction of spelling errors is used;

b. The effectiveness is evaluated by comparing with existing transliteration methods.

The main approaches to automating orthographic transliteration in the study include:

a. A rule-based approach - creating traditional transliteration rules and applying them algorithmically;

b. A statistical approach - conducting statistical analysis on texts to identify the most common transliteration changes;

c. Phonetic transliteration algorithms - obtaining improved transliteration results by taking into account the phonetic relationship between letters.

The study used the Python programming language and its libraries such as TensorFlow, PyTorch,

OpenNLP, spaCy, Pandas, NumPy, Django.

The research was carried out in the following stages:

a. Data collection – a corpus of texts written in Uzbek was formed. It included official documents, scientific articles and general texts;

b. Data pre-processing – data in Cyrillic and Latin alphabets was cleaned, encoding errors were corrected;

c. Preparation of transliteration models, rule-based model (for large changes), statistical analysis (detection and correction of common transliteration errors);

d. Model testing – each model was tested on a special test set and the results were compared;

e. Evaluation and analysis – the effectiveness of the models was measured using indicators such as F1-score (precision and recall), BLEU (Bilingual Evaluation Understudy), CER (Character Error Rate) and WER (Word Error Rate).

The research was carried out using the following software and hardware:

a. Software (Python 3.9) and its libraries tensorflow, pytorch (for neural networks), opennlp, spacy (for language processing), Pandas, numpy (for data analysis), Django (for creating a Web interface);

b. Equipment: NVIDIA RTX 3090 GPU (for Deep Learning model); 32 GB RAM, Intel Core i9-12900K processor; Cloud Computing (Google Colab, AWS).

Thus, the study developed a transliteration model based on the Uzbek text corpus, in which data collection was carried out as follows:

a. Official texts and documents - official announcements and scientific articles on the Uzbek language were downloaded;

b. Data collection from the Internet - texts from Uzbek web pages and blog posts were collected;

c. Annotation process - data was manually checked for transliteration and saved in the correct format;

d. Data cleaning and normalization - duplicates, spelling errors, and incorrect transliterations were eliminated;

e. Model evaluation and analysis of results - the accuracy and effectiveness of each model's results were assessed using special metrics.

The following results were achieved in terms of data volume and content:

a. More than 500 thousand words were collected for the study;

b. Texts were divided into specific categories (scientific articles - 35%; official documents and laws - 25%; blog posts and texts from the internet - 40%, etc.).

Based on the analysis of the collected data, the following results were obtained:

a. The ratio of Cyrillic-Latin letters was 60:40;

b. The most common transliteration errors occurred in the following cases: “Қ” → “K” or “Q” (incorrect transliteration cases); “Ғ” → “Г‘” or “Gh”; “Ҳ” → “H” or “X”; errors in hyphenation and syllable separation;

c. As a result of statistical analysis, the 100 most common words and their variants were identified.

The following were used as criteria for evaluating the model results:

a. F1-score – the model results were measured in terms of the balance between precision and recall;

b. BLEU (Bilingual Evaluation Understudy) – the model results were measured in terms of bilingual evaluation;

c. CER (Character Error Rate) – the model results were measured in terms of the error rate in characters;

d. WER (Word Error Rate) – the model results were measured in terms of the error rate in words and the results were as follows:

Precision: 0.87 points, Recall: 0.83 points, F1-score: 0.85 points (Figure 1).

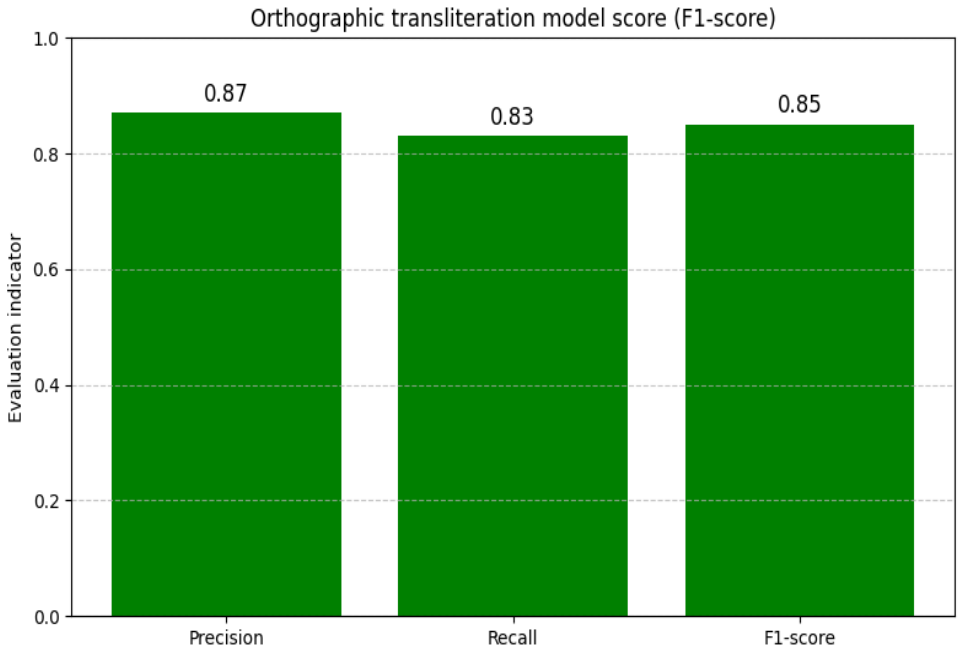

Figure 1. Orthographic transliteration model score (F1-score – max: 1.0 points, min: 0.50 points).

The above chart graphically depicts the Precision, Recall, and F1-score of the model. All values are above 0.80, indicating that the model is performing well. In particular, F1-score = 0.85 is a very positive result.

The diagram in Figure 2 shows the evaluation metrics (BLEU, CER, WER) of the orthographic transliteration model. Based on the results returned by each metric, it can be concluded that this model works well (the model was also compared with human transliterated texts to identify differences).

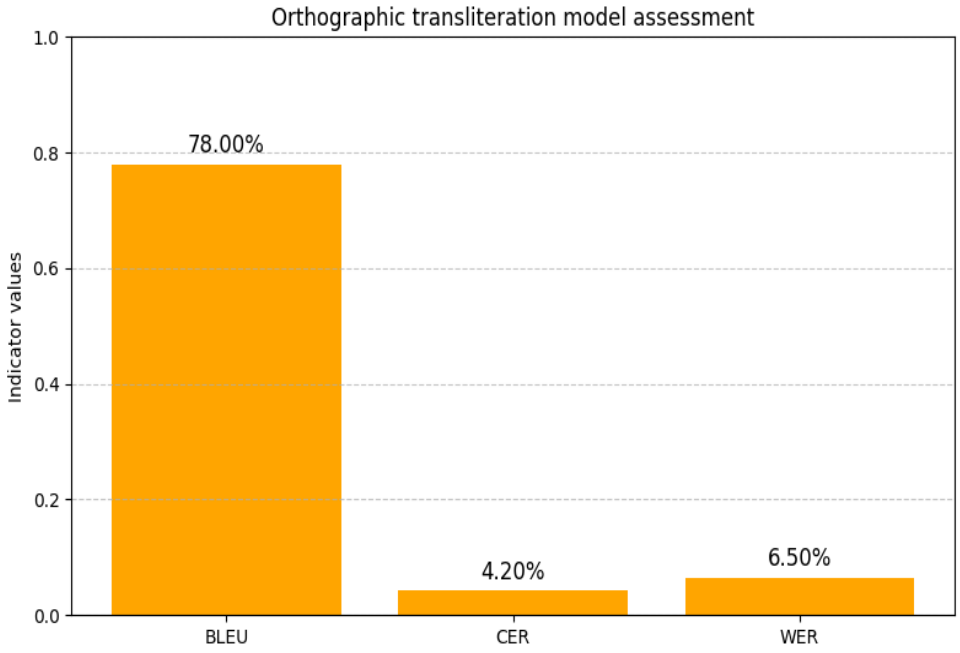

Figure 2. Orthographic transliteration model score (BLEU – min: 0% (poor), max: 100% (excellent), CER – min: 40% and above (poor), max: 0% (excellent), WER – min: 60% and above (poor), max: 0% (excellent)).

3. Results and Discussion

Orthographic transliteration is the conversion of text from one writing system to another, based on certain rules.

For example, in the Latin-Cyrillic and Cyrillic-Latin directions, transliteration is carried out based on phonetic, morphological and formal changes as follows:

Latin: “mo‘jiza (miracle)” → Cyrillic: “мўъжиза (miracle)”

Cyrillic: “мўъжиза (miracle)” → Latin: “mo‘jiza (miracle)”

Computer modeling of this process means formalizing and algorithmizing the translation process.

Linguistic basis: basic units used in transliteration (grapheme - letters of the alphabet (A, B, O‘, G‘, Sh, Ch, ...); orthographic rules - the interaction of letters (for example: sh, ch, o‘, g‘, ng - one unit); context - the characters before and after the letter.

Graphemic representations:

a. Transliteration function from Latin to Cyrillic: $T : L^* \to K^*$;

b. Transliteration function from Cyrillic to Latin: $T : K^* \to L^*$;

c. From Latin to Cyrillic – $R = \{(l_i, k_j)\}$, $i = \overline{1, n}$, $j = \overline{1, m}$;

d. From Cyrillic to Latin – $R = \{(k_j, l_i)\}$, $i = \overline{1, n}$, $j = \overline{1, m}$.

Here: T is a rule-based matching function for each grapheme (or group of graphemes); $L = (l_1, l_2, ..., l_n)$ – a set of letters and symbols that make up the Latin alphabet; L^* – a word(s) made up of Latin letters; $K = (k_1, k_2, ..., k_m)$ – a set of letters and symbols that make up the Cyrillic alphabet; K^* – a word(s) made up of Cyrillic letters; R is a relation that represents the connection

between the elements of the two sets.

For example, the following expressions can be practically written for the transliteration function in the process of converting from Latin to Cyrillic or from Cyrillic to Latin:

$$T("j") = "ж";$$

$$T("м") = "m"$$

Also, the set of rules that determine the transliteration method is expressed as follows:

$R = \{(l_i, k_j) | T(l_i) = k_j\}$ (Latin to Cyrillic);

$R = \{(k_j, l_i) | T(k_j) = l_i\}$ (Cyrillic to Latin).

Algorithmization process (in Latin-Cyrillic direction):

a. Text is entered. The following expression is written for the entered text:

$$Text = (L_1^*, L_2^*, ..., L_n^*).$$

b. Digraphs (i.e. letter combinations or chains of symbols) in words are treated as separate coded symbols, for example:

[g‘] = [g] + [‘],

[o‘] = [o] + [‘],

[sh] = [s] + [h],

[ch] = [c] + [h],

[ng] = [n] + [g].

c. For each unit:

latin_letter = T(cyrillic_unit);

d. Result:

$$Text' = (K_1^*, K_2^*, ..., K_m^*)$$

Transliteration works independently for each character (or unigram) and is expressed as follows:

$$Text' = \sum_{i=1}^{n} T(c_i | c_{i-1} c_{i+1}).$$

Where: c_i – current letter (or unit); $c_{i-1} c_{i+1}$ – context letters; T – rule-based matching (turning) function (sometimes context-dependent).

If we look at it from the point of view of the possibilities of programming languages, then we have to work with the following extended models, where the statistical model - each turn is selected based on probability, and the neural model represents the following complex function:

$$output = decoder(encode(input))$$

It is worth noting that the above models can be worked with using Python programming language libraries such as TensorFlow, PyTorch, OpenNLP, spaCy, Pandas, and NumPy.

Thus, the errors that occur (or may occur) during the transliteration process can be:

a. Phonetic similarity errors: transliterations that do not correspond to the sounds of the Uzbek language;

b. Orthographic errors: incorrectly transliterated words;

c. Rule exceptions: words that should be transliterated based on a special rule, but give an incorrect result.

A brief example of the result of orthographic transliteration through the Web application is given below, see Figure 3.

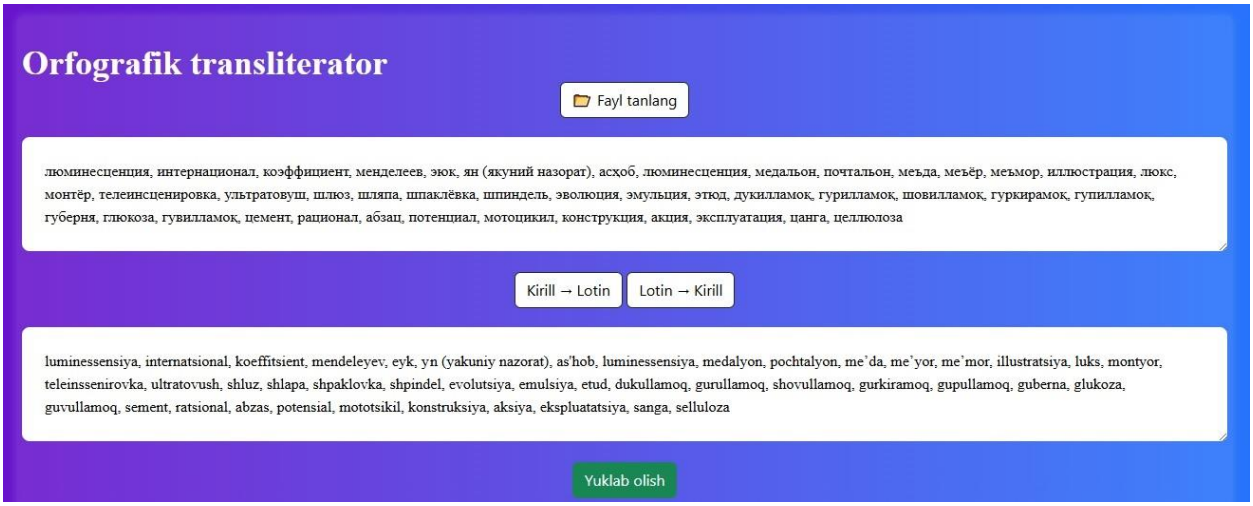

Figure 3. Web application for orthographic transliteration.

4. Conclusion

The results of our study have led to a number of important conclusions in the field of computer modeling of orthographic transliteration, and these results can be analyzed according to the following aspects:

a. This study has developed models that allow for accurate and error-free transliteration of Uzbek texts;

b. When comparing the results of the study with previously implemented methods, it was found that the new models improved the quality of transliteration;

c. Although the proposed models gave accurate and effective results, it is worth noting that there are a number of limitations, namely:

- in some cases, there is a possibility that words can be transliterated in several ways;
- errors may occur during transliteration in long texts.

Based on the results of the research, it is recommended to carry out work in the following areas:

a. Machine Learning models depend on a large, well-defined corpus, and for its effective use, the corpus should be enriched;

b. The transliteration model based on neural networks should be improved;

c. It is advisable to integrate natural language processing technologies;

d. It is necessary to develop algorithms for semantic-syntactic analysis of one-to-one correspondence between words in Cyrillic and Latin scripts;

e. It is necessary to develop and apply algorithms for automatic detection and correction of spelling errors.

References

[1] K. Knight and J. Graehl, "Machine transliteration," in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, Madrid, July 1997, pp. 128–135. [Online]. Available: https://doi.org/10.3115/976909.979634

[2] A. Finch and E. Sumita, "Phrase-based machine transliteration," in *Proceedings of the Workshop on Technologies and Corpora for Asia-Pacific Speech Translation (TCAST)*, 2008.

[3] B. Mansurov and A. Mansurov, "Uzbek cyrillic-latin-cyrillic machine transliteration," *arXiv preprint*, arXiv:2101.05162, 2021.

[4] U. Salaev, E. Kuriyozov, and C. Gómez-Rodríguez, "A machine transliteration tool between Uzbek alphabets," *arXiv preprint*, arXiv:2205.09578, 2022.

[5] T. Togayev, G. Tavaldiyeva, and M. Akromova, *Spelling dictionary of the Uzbek language in Cyrillic and Latin alphabets*. Tashkent: Editorial Office of the “Sharq” Publishing and Printing Joint-Stock Company, 1999. (In Uzbek).