Formalization of Morphological Rules for Kazakh Nouns in the New Latin Alphabet

Lena Zhetkenbay[1,*], Altynbek Sharipbay[2], Bibigul Razakhova[3], Gulmira Bekmanova[4], Alibek Barlybayev[5], Aizhan Nazyrova[6,*], Banu Yergesh[7]

[1,2,3,4,5,6,7]*Department of Artificial Intelligence Technologies, Faculty of Information Technologies, L.N. Gumilyov Eurasian National University, Pushkina 11, Astana 010008, Kazakhstan*

(Received: January 20, 2025; Revised: February 28, 2025; Accepted: May 30, 2025; Available online: July 13, 2025)

Abstract

This study presents a hybrid computational model for formalizing and predicting morphological inflections of Kazakh nouns written in the new Latin alphabet. The motivation stems from limitations in previous systems based on Cyrillic orthography, which often misrepresented key phonological features such as vowel harmony and consonant assimilation. The main objective is to develop a linguistically informed and computationally efficient system to support Natural Language Processing (NLP) for Kazakh during its transition to Latin script. The methodology combines rule-based grammar formalization with a machine learning approach, specifically a Bayesian Regulation Backpropagation Neural Network (BR-BPNN). A manually curated dataset of 1,000 Latin-script Kazakh nouns was annotated for various morphological forms. Each word was encoded at the character level using a custom dictionary (kazlat_dict), capturing the final four letters as feature vectors. Formal logic and regular expressions were used to model morphological rules such as pluralization and case endings, incorporating vowel harmony, consonant softness, and sonority. These rules provided the training labels for the BR-BPNN model. The trained model achieved 91.5% accuracy, 89.4% precision, and a correlation coefficient (R) above 0.98, confirming the effectiveness of the hybrid system. A user interface prototype was developed to demonstrate practical utility, enabling users to input root nouns and receive suffix predictions with confidence scores and linguistic explanations. The novelty of this work lies in integrating linguistic theory with machine learning for a low-resource Turkic language. It offers a foundation for intelligent Kazakh language tools including spell checkers, grammar correctors, and educational platforms. Future work will extend the system to other parts of speech and explore contextual modeling to improve handling of ambiguous or irregular forms.

Keywords: Alphabet, Sound System, Nouns, Conjunctions, Kazakh Language, Suffixes, Metalanguage, Morphological Rules, Natural Language Processing, Formal Model

1. Introduction

The Kazakh language stands as one of the historically rich members of the Turkic language family. Its structure, formed through centuries of development across the vast steppes of Central Asia, reflects deep linguistic traditions shared with other Turkic tongues [1]. One of its most defining characteristics is its agglutinative morphology, where words evolve through the consistent attachment of suffixes, each adding a new grammatical or semantic function. This structural regularity, especially the phonological harmony that governs suffixation, opens promising pathways for formal modeling and computational treatment. As Kazakhstan enters a new linguistic era by transitioning to a Latin-based alphabet, the opportunity arises to align the orthographic system with the true phonetic identity of the Kazakh language. This change, far more than symbolic, aims to modernize the language’s technological infrastructure and establish closer connections with Kazakh-speaking communities beyond national borders. For many abroad, Latin-script Kazakh has become a practical norm, especially in digital communication spaces [2].

In response to this shift, developing formal linguistic models grounded in Latin orthography becomes a timely and necessary endeavor. Previous modeling efforts, rooted in ontological frameworks [3], [4], [5], offered valuable insights into the structure of Kazakh grammar. Yet, practical deployment exposed fundamental inconsistencies. These stemmed mainly from the limitations of the Cyrillic-based system, which included symbols for non-Kazakh sounds and failed to reflect the language’s phonological logic. Recent research in phonetics and language planning has shed light on the

***Corresponding author: Aizhan Nazyrova (nazyrova_aye_1@enu.kz)**

DOI: https://doi.org/10.47738/jads.v6i2.820

shortcomings of both the Cyrillic and earlier Latin proposals. The new alphabet, influenced by the Turkish model, resolves many of these issues by offering a closer match to the language's phoneme inventory. With a consistent representation of vowels and semivowels, this script sets the stage for rule systems that better respect Kazakh's linguistic nature [2].

Formal models built on this foundation have found application in socially relevant domains, including the automatic analysis of political texts. As part of the AP19679847 project on Kazakh political discourse, tools based on Latin-script morphology are being developed to process and interpret posts by diaspora communities. These messages often address key societal developments, ranging from constitutional reforms to shifts in public opinion. For example:

"Qazaqstanda sayasi reforma jürgizilip jatır. Bul jaña konstituciyalık özgerister halıqtıñ ükimetke degen senimin arttırıuı mümkin." ("Kazakhstan is undergoing political reforms. The recent constitutional amendments may enhance public trust in the government."). Expressions like this, written outside Kazakhstan but deeply tied to its political life, require linguistic models that can parse structure and extract meaning with precision. Given the diasporic variation in pronunciation and spelling, the models must also accommodate phonetic flexibility without sacrificing formal rigor [2].

To achieve this, a clear meta-linguistic framework is essential. The UniTurk system serves this purpose, offering a standardized set of symbols and categories for representing Turkic grammatical phenomena [6]. Through UniTurk, this research encodes noun morphology using formal logic, regular expressions, and inference rules, designed not only to reflect theoretical structure but also to support practical applications in computational environments.

2. Literature Review

Digital transformation has significantly influenced the way national languages are studied and processed. In countries such as the United States, the United Kingdom, Russia, Japan, and China, the development and teaching of native languages increasingly rely on computational methods. These include the implementation of intelligent tutoring systems, machine translation engines, electronic grammar checkers, and platforms for speech recognition and synthesis. The linguistic rules of these languages are encoded in formal systems and embedded into software applications, enabling automated processing at various levels- phonetic, morphological, syntactic, and semantic.

Kazakh linguistics has also followed this trend. Over the past decades, numerous research projects have focused on analyzing the morphological and syntactic structures of the Kazakh language, automating translation tasks between Kazakh and other languages, and developing tools for the analysis and generation of both written and spoken texts. The results of these initiatives have been published in high-impact journals and conference proceedings indexed in Scopus and Web of Science [7], [8], [9], [10], [11], [12]. These contributions have led to the creation of electronic grammar reference books, morphological processors capable of recognizing and generating Kazakh word forms, multilingual thesauri, and educational platforms that incorporate question-answering modules and intelligent knowledge assessment tools.

In parallel, other research groups have also developed rule-based morphological analyzers for Kazakh, focusing on finite-state transducers and formal grammar models [13], [14], [15]. One of the prominent directions involves constructing ontological models to represent the grammatical framework of the language [16], [17]. These ontologies aim to capture dependencies among grammatical features and support the generation of well-formed expressions. Despite their value, the practical implementation of such systems often reveals internal contradictions, particularly when dealing with edge cases or phonologically irregular forms.

A key limitation in most previous models is their reliance on the Cyrillic script, which introduces complications due to its inclusion of phonemes not native to the Kazakh language. These discrepancies affect the accuracy of morphological generation and recognition systems, especially in cases involving vowel harmony or phoneme assimilation. This creates a barrier for developing rule sets that align with the natural phonological patterns of Kazakh. Addressing this requires a shift to a writing system that better reflects the native phonetic inventory.

Our research proposes a model that addresses this need by formalizing noun inflection rules in Latin script and evaluating them using neural models trained on Kazakh morphological patterns. Unlike prior approaches, the proposed method explicitly accounts for phonological harmony and includes error-prone scenarios that allow the identification of weak points in existing rule systems. This integrated approach builds upon earlier ontological models, enhancing their performance through data-driven validation and linguistic consistency.

3. Methodology

This study implements a hybrid approach that combines the formalization of morphological rules with neural modeling to accurately capture the inflectional behavior of Kazakh nouns written in the new Latin-based script. The step-by-step research process is illustrated in figure 1.

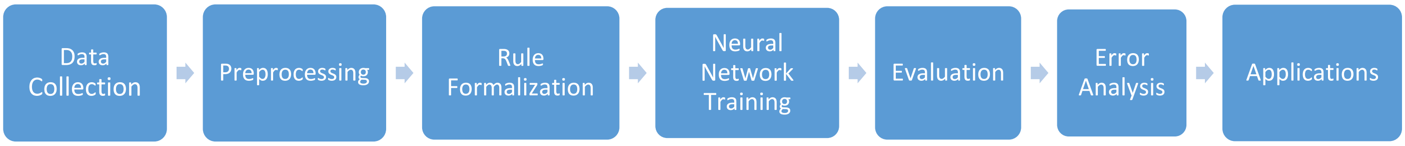


Figure 1. Architecture of the BR-BPNN Model for Kazakh Suffix Prediction Based on Character-Level Input

The study began by compiling a dataset of 1 000 manually selected Kazakh nouns written in Latin script to represent a diverse range of phonological and morphological structures. These words were carefully chosen to include both regular and irregular forms, particularly in relation to vowel harmony and suffix attachment. Each word was transformed into a numerical representation by encoding its last four characters using a predefined dictionary (kazlat_dict), enabling structured input suitable for training a neural network. To guide the morphological transformations, rules were formalized using logical expressions and regular expressions that accounted for phonological features such as vowel harmony, sonority, and consonant type. These rules were expressed in production form and verified through linguistic analysis to ensure their validity. A BR-BPNN was developed to model inflectional patterns, trained on input-output pairs that mapped root forms to their correctly inflected variants. The performance of the model was evaluated using standard metrics such as Mean Squared Error (MSE) and the correlation coefficient (R), with the system demonstrating high predictive accuracy (R > 0.98 on test data), thereby validating the effectiveness of the hybrid rule-based and machine learning approach. Despite its success, the system struggled with root words that featured complex phonological environments or allowed multiple valid suffixes, revealing limitations in the dataset and indicating a potential need for context-sensitive modeling strategies. The proposed methodology has promising applications in language processing tools such as spell checkers, grammar correction modules, and educational software tailored for Kazakh texts in the Latin script. Furthermore, it can facilitate linguistic analysis in digital environments where precise morphological processing is critical.

3.1. Dataset Description

The dataset used in this study comprises 1000 manually annotated noun forms in the Kazakh language, transcribed using the standardized Latin-based alphabet. Each instance includes a base noun and its correctly inflected form, generated according to the morphological rules of the Kazakh language. The dataset encompasses both singular and plural forms, reflecting variation in phonological features such as vowel harmony and consonant assimilation.

The average word length is approximately 6.5 characters, with examples ranging from short words (e.g., *su* → *suğa* (water → to the water)) to longer forms (e.g., *kitaptarmen* (with the books)). The corpus comprises approximately 800 unique noun roots and encompasses 35 distinct suffix types, which serve as classification labels. Suffix frequency is unevenly distributed: common forms such as "*qa*", "*ke*", and "*men*" (to / into / toward) appear 100, 95, and 87 times respectively, while less frequent suffixes like "*şa*" and "*nen*" (to / onto) occur fewer than 10 times. This imbalance was taken into account when evaluating classification performance.

All data were preprocessed by converting words to lowercase and removing special characters. Tokenization was performed at the character level, with resulting sequences ranging from 5 to 15 characters. Label encoding was applied to suffixes for use in supervised learning tasks. Overall, the dataset captures key aspects of Kazakh noun morphology within the constraints of the Latin alphabet, making it suitable for tasks involving morphological classification and generation.

3.2. Model Training and Evaluation

The BR-BPNN model was trained using 80% of the data, with the remaining 20% allocated for validation—the training process aimed to identify optimal weights that minimize the prediction error. The model's performance was assessed using accuracy, MSE, and the R. These metrics allowed for the evaluation of both classification accuracy and the strength of the relationship between predicted and actual suffix forms.

3.3. Rule-Based Component

To encode regular morphological behavior, a formal rule engine was developed using pattern-matching logic. Each rule was expressed in regular expression syntax, allowing for the capture of systematic alternations such as plural suffixes *(-лар, -лер, -дар, -дер, -тар, -тер*). These rules matched stem-final vowel classes and consonant patterns, enabling affix selection by Kazakh vowel harmony. Edge cases and rule violations were not discarded; instead, they were embedded into the system as conditional branches or exceptions, enhancing linguistic coverage.

3.4.Vectorization and Normalization Details

To begin the vectorization process, each character in the Kazakh root words written in Latin script is assigned a unique numerical Identifier (ID) using a predefined dictionary. This dictionary contains 34 characters, covering all phonologically relevant Latin-based Kazakh letters. It serves as the foundation for converting text into numerical vectors. The complete mapping of characters to IDs is presented in table 1.

Table 1. Latin-Based Kazakh Character Dictionary

Letter	Code	Letter	Code
'a'	'1'	'n'	'18'
'ä'	'2'	'ñ'	'19'
'b'	'3'	'o'	'20'
'c'	'4'	'ö'	'21'
'ç'	'5'	'p'	'22'
'd'	'6'	'q'	'23'
'e'	'7'	'r'	'24'
'f'	'8'	's'	'25'
'g'	'9'	'ş'	'26'
'ğ'	'10'	't'	'27'
'h'	'11'	'u'	'28'
'i'	'12'	'ü'	'29'
'ı'	'13'	'v'	'30'
'j'	'14'	'x'	'31'
'k'	'15'	'y'	'32'
'l'	'16'	'z'	'33'
'm'	'17'	'w'	'34'

To ensure equal contribution of all features and stabilize neural network training, the raw vectors were normalized using the following formula:

$$Normalized\ value = \frac{Character\ Code}{34} \quad (1)$$

This transformation maps values to the range of [0, 1]. The results of this process are shown in table 2.

Each Latin-based Kazakh root word was truncated to its final four characters to capture morphological and phonological cues relevant to suffix prediction. These letters were then converted to numerical IDs based on the dictionary provided in table 2. For instance, the word balalar ends with the sequence a-l-a-r, which is mapped to the raw vector [1], [16], [24].

Table 2. Sample Vectorization And Normalization

Word	Last four letters	Raw vector	Normalized vector
balalar – children	a-l-a-r	[1, 16, 1, 24]	[0.03, 0.48, 0.03, 0.73]
köşeler – corners	e-l-e-r	[7, 16, 7, 24]	[0.21, 0.48, 0.21, 0.73]
kitaptar – books	p-t-a-r	[22, 27, 1, 24]	[0.65, 0.79, 0.03, 0.71]

This preprocessing pipeline, from truncation to character mapping (table 1) and normalized vector representation (table 2), provides consistent and interpretable inputs for the neural model, while preserving essential phonological cues in Kazakh morphology.

4. Changing the Alphabet of the Kazakh Language

In all spheres of human activity, intelligent technologies are being used for written and oral communication in the languages of sovereign countries. The creation of such technologies requires voluminous and complex scientific and practical work on creating ontologies of subject areas, formalizing grammatical rules of natural languages, and developing language processors to analyze and generate written units. However, during the training of such technologies in the Kazakh language, it was noticed that there are contradictions in morphological rules [2]. To eliminate these contradictions, the Kazakh language's sound system is planned to be standardized, and a new alphabet based on the Latin alphabet of the Turkish language will be adopted in place of the existing one.

The current Cyrillic-based alphabet consists of 42 letters, 15 of which (*Аа, Әә, Ее, Ёё, Ии, Оо, Өө, Уу, Ұұ, Үү, Ыы, Іі, Ээ, Юю, Яя*) denote vowel phonemes and 25 (*Бб, Вв, Гг, Ғғ, Дд, Жж, Зз, Йй, Кк, Ққ, Лл, Мм, Нн, Ңң, Пп, Рр, Сс, Тт, Фф, Хх, Һһ, Цц, Чч, Шш, Щщ*) denote consonant phonemes. Two letters (*Ьь, Ъъ*) are used to indicate the softness and hardness of consonants. The letters *Ёё, Ээ, Юю, Яя, Щщ, Ьь, Ъъ* denote sounds that are not related to the Kazakh language and are not used in affixes (suffixes and endings) of words. Therefore, they do not need to be included in the new alphabet. The letters *Ии* and *Уу* denote the vowel phonemes of the Russian language (*и*) – [i] and (*у*) – [u] (here in parentheses are the names of phonemes, and in square brackets are the symbols of these sounds in the International Phonetic Alphabet [18]). The letters *Ии* and *Уу* are involved in the affixes of Kazakh words and give rise to contradictions in morphological rules:

4.1. Law of Synharmonism [19]

In Kazakh root words, vowels must alternate with consonants, and only soft or only hard vowels should be used in recording, but when using these letters, this condition is violated. For example, in the following words, "*қиа* (cliff)", "*иә* (yes)", "*ауа* (air)", "*әуе* (sky)", "*уақыт* (time)", "*уәде* (promise)", "*кие* (shrine)", "*қиуа* (away)", "*қия* (obliquely)". "*саяхат* (journey)", "*сүю* (love)", "*сұю* (liquefy)" letters of vowels occur in a row without regard to their softness and hardness;

4.2. Possessive Endings, 3 People [20]

If the stem of a word (root or root + suffix) ends with a hard (or soft) vowel, then a possessive ending in the third person "сы" (or "сі") is added to it. For example, "*ана + сы* (mother)", "*әже + сі* (grandmother)" [2]. If the hard (or soft) stem of a word ends in a consonant, then a possessive ending in the third person "*ы*" (or "і") is added to it. For example, "*отан + ы* (homeland)", and "*ел + і* (country)".

But when using the letters *Ii* and *Uu*, which denote vowels of the Russian language, in the record of the stem of the word, these rules are violated. For example, "*би + і* (dance)", "*ми + ы* (brain)", "*ту + ы* (flag)" and "*гу + і* (hum)" are written instead of writing according to the rule "*би + сі*", "*ми+сы*, "*ту+сы*" and "*гу+сі*", which have no meaning in the Kazakh language [2].

These examples demonstrate the fallacy of the current Kazakh alphabet, which is based on the Cyrillic alphabet. Therefore, reform of the Kazakh language is required. Thus, 33 phonemes will participate in the reform of the translation of Kazakh script into Latin script, of which 9 are vowel phonemes (а), (ә), (е), (о), (ө), (ұ), (ү), (ы), (і) and 24 consonant phonemes (б), (в), (г), (ғ), (д), (ж), (з), (й), (к), (қ), (л), (м), (н), (ң), (п), (р), (с), (т), (w), (ц), (ф), (х), (ч), (ш). The reform will also include semivowels (i) and (у) as allophones of vowel phonemes (и) and (у), respectively [2].

The vowel sounds (и) and (у) borrowed from the Russian language, including in the sound system of the Kazakh language as allophones of the vowel phonemes (і) and (ұ), are denoted by letters used to designate the phonemes (і) and (ұ), respectively. For example, if the vowel phonemes (і) and (ұ) were denoted by Latin letters Ii and Uu, then it would not be challenging to write and read such terms as internet, institute, university, or supremum. If borrowed vowels (и) and (у) are denoted by separate letters in the new alphabet, then the contradictions mentioned in the 2nd paragraph will inevitably appear. When determining the composition of the new Kazakh alphabet, phonemes for which there are adequate Latin letters are denoted by these letters without diacritics and digraphs [2].

Phonemes for which there are no adequate Latin letters will create problems. These include soft vowel phonemes (ə), (ө), and (ү), as well as consonant phonemes (ғ), (н), (ц), (h), and (ш). To address these issues, two options are presented [2]. A new Kazakh alphabet based on the Turkish alphabet: Soft vowel phonemes (ə), (ө), (ү) are denoted by the Latin letters Aa, Oo, and Uu with a superscript “two dots” diacritic Ä ä, Ö ö, Ü ü, respectively; Soft vowel phoneme (і) is marked with the letter Iı with a superscript diacritic “one dot” İ i, and the hard vowel phoneme (ы) consonant with it are marked with the letter without the diacritic I ı; Consonant phonemes (ғ) and (н) are denoted by the Latin letters Gg and Nan with accented diacritic signs "brevis" Ğğ and Ňň, respectively; Consonant phonemes (ч) and (ш) are denoted by Latin letters with subscript diacritic signs "cedilla" Çç and Ş ş, respectively.

In the new alphabet of the Kazakh language, it is recommended to designate with the help of the Latin letter Xx, not one phoneme, but a combination of consonant phonemes (k) and (s). The use of this letter does not create any contradictions in the Kazakh script and allows writing many international scientific, technical, and other terms in the original English or close to it. For example, axis – *ось*, axiom – *аксиома*, accelerate – *акселерат*, box – *бокс*, exel – *эксель*, experience – *опыт*, expert – *эксперт*, export – *экспорт*, context – *контекст*, maximum – *максимум*, mixer – *миксер*, Oxford – *Оксфорд*, taxi – *такси*, xerox – *ксерокс*. Table 3 presents the Latin version of the Kazakh alphabet, which is based on the Turkish alphabet.

Table 3. The Version of the Kazakh Alphabet is Based on the Turkish Alphabet

Latin	Cyrillic	IPA	Latin	Cyrillic	IPA
A a	А а	[ɑ]	N n	Н н	[n]
Ää	Ә ә	[æ]	Ñ ñ	Ң ң	[ŋ]
B b	Б б	[b]	O o	О о	[ɔ]
C c	Ц ц	[tc]	Ö ö	Ө ө	[ɵ]
Çç	Ч ч	[tʃ]	P p	П п	[p]
D d	Д д	[d]	Q q	Қ қ	[q]
E e	Е е	[e]	R r	Р р	[r]
F f	Ф ф	[f]	S s	С с	[s]
G g	Г г	[g]	Ş ş	Ш ш	[ʃ]
Ğ ğ	Ғ ғ	[ɣ]	T t	Т т	[t]
H h	Х х	[h]	U u	Ұ ұ	[ʊ u]
I ı	Ы ы	[ɯ]	Ü ü	Ү ү	[y]
İ i	І і	[ɪ, i]	V v	В в	[v]
J j	Ж ж	[ʒ]	W w		[w]
K k	К к	[k]	X x		[ks]
L l	Л л	[l]	Y y	Й й	[y]
M m	М м	[m]	Z z	З з	[z]

5. Formalization of Rules for Forming Nouns in the Kazakh Language

5.1. Symbols of the Elements of the Kazakh Language

To formalize the morphological rules of the Kazakh language, the following designations are acceptable:

N is a set of nouns; $\mathrm{N_j}$ is a set of singular nouns; $\mathrm{NPlur} = \{\mathrm{lar}, \mathrm{ler}, \mathrm{dar}, \mathrm{der}, \mathrm{tar}, \mathrm{ter}\}$ is a set of plural nouns; Nom is a set of nouns in the nominative case; NGen is a set of nouns with a genitive case; NDir is a set of nouns in the subjective case (uses the preposition 'to'); NAcc is a set of nouns in the objective case; NLoc is a set of nouns in the common case (answers the question ‘where’); NAbl is a set of nouns in the common case (uses the prepositions 'by' and 'with'); NInst is a set of nouns in the common case (uses prepositions 'about', 'of'); Adj is a set of the main adjectives in the Kazakh language; AdjEnhanDeg is a set of the superlative degree of adjectives of the Kazakh language; AdjCompDeg is a set of the comparative degrees of adjectives of the Kazakh language; Plur is a set of plurals of nouns; Cases are a set of case endings of nouns; Nom is a set of nominative cases; Gen is a set of genitive cases; Dir is a set of subjective cases; Acc is a set of objective cases; Loc is a set of nouns in the common case (answers the question 'where'); Abl is a set of nouns in the common case (uses the prepositions 'by' and 'with'); Inst is a set of nouns in the common case (uses prepositions ‘about’, ‘of’); Values α, β, γ, δ, φ are formed from Kazakh letters by concatenation '·' variables that are arrays of length greater than 1; ∪ Is a combination operation.

Any part of speech in the Kazakh language is divided into syllables, which include one vowel, and depending on whether the last syllable is thick or thin, thick or thin endings are added. The last syllable in the given part of speech is formally written in the language of mathematical logic as follows:

For thick syllables:

$$S_h(\beta\cdot\gamma) \rightleftharpoons \beta\cdot\gamma = y\cdot z, \exists\xi(\xi \in V_h \& \, \xi \subseteq z), \forall\zeta(\zeta \neq \xi \; \& \, \zeta \subseteq z \; \rightarrow \; \zeta \notin V) \tag{2}$$

for thin syllables:

$$S_s(\beta\cdot\gamma) \rightleftharpoons \beta\cdot\gamma = y\cdot z, \exists\xi(\xi \in V_s \& \, \xi \subseteq z), \forall\zeta(\zeta \neq \xi \; \& \, \zeta \subseteq z \; \rightarrow \; \zeta \notin V) \tag{3}$$

S(x) is a predicate whose value is “True” or “False”, x is a given phrase, ⇌ is a symbol meaning “Definable”, z is the last syllable of the words, y is a residue that does not include the previous syllable, “·” is concatenation (attachment) the operation, “&” is conjunction (and) operation, “⊆” is a relation that the substring on the left is included in the substring on the right. The main words of speech in the Kazakh language are Nouns, Adjectives, Numerals, Pronouns, Verbs, Adverbs and etc. al. [21].

5.2. Formalize Rules for Adding Suffixes to Nouns

A noun is a part of speech that denotes an object and answers the questions "Who?" and “What”. The conjunction connects two words. It has four types: plural ending, case ending, possessive ending, and personal ending. Plural. There are six types of plurals: *lar, ler, dar, der, tar,* and *ter*. The plural suffix is added to the word (noun) according to the law of harmony and indicates the plurality of things. For example: *bala·lar*(children), *äke·ler*(fathers), *qız·dar* (daughters), *jiyen·der*(nieces), *kitap·tar*(books), *mektep·ter* (schools), etc. In natural speech, words are made up of sounds, and in writing, they are represented by letters. Phrases and sentences are created by combining words, and the following types of special punctuation marks are placed on them (table 4):

Table 4. Special Punctuation Marks

Dot	Ellipsis	Colon	Comma	semicolon	minus	hyphen	quotation mark	question mark	exclamation mark
.	...	:	,	;	–	-	«»	?	!

To clarify the rules for users, let's outline the connection of plural and adverbial endings in the Kazakh language as simply as possible, and consider formalizing them in the form of production rules. To formalize the rules for conjugating nouns in the Kazakh language, we first introduce the following characteristics. Using this notation, we write formal derivation rules in the form of a sequence$\frac{A}{B}$, where A is an antecedent, and B is a consequent, according to the rules of the Kazakh language. We are looking at a collection of sounds and applications. Morphological rules in the Kazakh language also refer to syllables. Since vowels define the syllables, we represent syllables by vowels to formalize the rules. For example, “ma” has the hard vowel sound «a», so the last syllable is considered hard to determine in the word “alma”. Such determination is necessary in cases where the words are neither a hard vowel sound nor a sonorous vowel sound. For example, books, wheat, Elaman, and others. Since suffixes are connected depending on sound harmony, we have divided the consonants into subgroups. Plural endings:

Rule 1. If a noun ends in a back vowel, the plural suffix 'lar' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta,\ \ \beta \in \{\mathrm{a}, \mathrm{ä}, \mathrm{ı}\ \}}{\delta \cdot lar = \varphi,\ \ \varphi \in NPlur} \tag{4}$$

Rule 2. If a noun ends in a front vowel, the plural suffix 'ler' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta,\ \ \beta \in \{\mathrm{e},\ \mathrm{i}\}}{\delta \cdot ler = \varphi,\ \ \varphi \in NPlur} \tag{5}$$

Rule 3. If the last syllable of the noun ends in a sonorous consonant $\{\mathrm{y}, \mathrm{w}, \mathrm{r}\}$, then the plural suffix 'lar' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma,\ S_h(\beta \cdot \gamma), \gamma \in \{y, w, r\}}{\delta \cdot lar = \varphi,\ \ \varphi \in NPlur} \tag{6}$$

Rule 4. If the last syllable of the noun is soft (front vowel) and ends in sonorant consonants $\{y, w, r\}$, then the plural suffix 'ler' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma,\ S_s(\beta \cdot \gamma), \gamma \in \{y, w, r\}}{\delta \cdot ler = \varphi,\ \ \varphi \in NPlur} \tag{7}$$

Rule 5. If the last syllable of a noun is hard (back vowel) and ends in a sonorant consonant $\{l, m, n, \mathrm{ñ}\}$, then the plural suffix 'dar' is added to the noun:

$$\frac{\mathrm{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{l, m, n, ñ\}}}{\mathrm{\delta \cdot dar = \varphi,\ \ \varphi \in NPlur}} \tag{8}$$

Rule 6. If the last syllable of a noun is soft (front vowel) and ends in the non-sonorant consonant l, then the plural suffix 'der' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma = l}{\delta \cdot der = \varphi,\ \ \varphi \in NPlur} \tag{9}$$

Rule 7. If the last syllable of the noun is hard (front vowel) and ends in non-sonorant consonants {j, z}, then the plural suffix 'dar' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{j, z\}}{\delta \cdot dar = \varphi,\ \ \varphi \in NPlur} \tag{10}$$

Rule 8. If the last syllable of a noun is soft (front vowel) and ends in non-sonorant consonants {j, z}, then the plural suffix 'der' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{j, z\}}{\delta \cdot der = \varphi, \ \ \varphi \in NPlur} \tag{11}$$

Rule 9. If the last syllable of a is hard (back vowel) and ends in non-sonorant consonants {b, v, g, d}, then the plural suffix 'tar' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\mathrm{b, v, g, d}\}}{\delta \cdot tar = \varphi, \ \ \varphi \in NPlur} \tag{12}$$

Rule 10. If the last syllable of is soft (front vowel) and ends in non-sonorant consonants {b, v, g}, then the plural suffix 'ter' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{b, v, g}\}}{\delta \cdot ter = \varphi, \ \ \varphi \in NPlur} \tag{13}$$

Rule 11. If the last syllable of a noun is hard (back vowel) and ends in voiceless sonorant consonants $\{q,\ p,\ s,\ t,\ f, ş\}$, then the plural suffix 'tar' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{q,\ p,\ s,\ t,\ f, ş\}}{\delta \cdot tar = \varphi, \ \ \varphi \in NPlur} \tag{14}$$

Rule 12. If the last syllable of a noun is soft (front vowel) and ends in voiceless sonorant consonants {k, p, s, t, f, ş}, then the plural suffix 'ter' is added to the noun:

$$\frac{\delta \in N_j, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\ k, p, s, t, f, ş\}}{\delta \cdot ter = \varphi, \ \ \varphi \in NPlur} \tag{15}$$

Case endings:

Rule 1. If a noun ends in a back vowel, then the genitive case ending ‘nıň’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \beta \in \{\mathrm{a, ä,\ ı\ }\}}{\delta \cdot n\text{ıň} = \varphi,\ \varphi \in NGen} \tag{16}$$

Rule 2. If a noun ends in a front vowel, then a genitive case ending hard ‘niň’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \beta \in \{\mathrm{e, i}\}}{\delta \cdot ni\text{ň} = \varphi,\ \varphi \in NGen} \tag{17}$$

Rule 3. If the last syllable of a noun is hard and ends in a vowel and ends in sonorant consonants (y, w, r, l), then the genitive case ending ‘dıň’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\mathrm{y, w, r, l}\}}{\delta \cdot d\text{ıň} = \varphi,\ \varphi \in NGen} \tag{18}$$

Rule 4. If the last syllable of a noun is a vowel and ends in a sonorous consonant (y, w, r, l), then the genitive ending ‘diň’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{y, w, r, l}\}}{\delta \cdot di\text{ň} = \varphi,\ \varphi \in NGen} \tag{19}$$

Rule 5. If the last syllable of a noun is hard and ends in a vowel and ends in non-sonorant consonants (j, z), then the genitive ending ‘dıň’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\text{j}, \text{z}\}}{\delta \cdot d\text{ıň} = \varphi,\ \varphi \in NGen} \tag{20}$$

Rule 6. If the last syllable of a noun is soft and ends in a vowel and ends in non-sonorant consonants (j, z), then the plosive ending ‘din’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\text{j}, \text{z}\}}{\delta \cdot di\text{ň} = \varphi,\ \varphi \in NGen} \tag{21}$$

Rule 7. If the last syllable of a noun is hard and ends in non-sonorant consonants (m, n, ň), then the genitive ending ‘nıň’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\text{m}, \text{n}, \text{ň}\}}{\delta \cdot n\text{ıň} = \varphi,\ \varphi \in NGen} \tag{22}$$

Rule 8. If the last syllable of a noun is soft and ends in sonorant consonants (m, n, ň), then the genitive case ending ‘niň’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\text{m}, \text{n}, \text{ň}\}}{\delta \cdot ni\text{ň} = \varphi,\ \varphi \in NGen} \tag{23}$$

Rule 9. If the last syllable of a noun is hard and ends in voiceless consonants ($p,\ s,\ t,\ f,$ ş) or non-sonorant consonants (b, v, g, d), then genitive case ending 'tıň' is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{b, v, g, d,\ p,\ s,\ t,\ f, \text{ş}\}}{\delta \cdot t\text{ıň} = \varphi,\ \varphi \in NGen} \tag{24}$$

Rule 10. If the last syllable of a noun is soft and ends in voiceless consonants (k, p, s, t, f, ş) or non-sonorant consonants (b, v, g, d), then the genitive case ending case 'tin':

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{b, v, g, d, k,\ p,\ s,\ t,\ f,\ \text{ş}\}}{\delta \cdot ti\text{ň} = \varphi,\ \varphi \in NGen} \tag{25}$$

Rule 11. If a noun ends in a back vowel, then the ending ‘a’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \beta \in \{\text{a}, \text{ä},\ \text{ı}\}}{\delta \cdot\ \text{ğa} = \varphi,\ \varphi \in NDir} \tag{26}$$

Rule 12. If a noun ends in a front vowel, then the dative case ending ‘ge’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \beta \in \{\text{e},\ \text{i}\}}{\delta \cdot \text{ge} = \varphi,\ \varphi \in NDir} \tag{27}$$

Rule 13. If the last syllable of a noun is hard and ends in non-sonorant consonants (j, z) or sonorant consonants (m, n, ň, y, w, r, l), then the dative case ending ‘ğa’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\text{j}, \text{z}, \text{m}, \text{n}, \text{ň}, y, w, r, l\}}{\delta \cdot\ \text{ğa} = \varphi,\ \varphi \in NDir} \tag{28}$$

Rule 14. If the last syllable of a noun is soft and ends in non-sonorant consonants (j, z) or in sonorant consonants (m, n, ň, y, w, r, l), then the dative case ending ‘ge’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{j, z, m, n, \text{ň}}, y, w, r, l\}}{\delta \cdot \mathrm{ge} = \varphi,\ \ \varphi \in NDir} \tag{29}$$

Rule 15. If the last syllable of a noun is hard and ends in non-sonorant consonants (b, v, g, d) or voiceless consonants (c, ç, f, k, q, p, s, ş, t, h), then the dative case ending 'qa' is added:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\mathrm{b, v, g, d}, c, \text{ç}, f, k, q, p, s, \text{ş}, t, h\}}{\delta \cdot qa = \varphi,\ \ \varphi \in NDir} \tag{30}$$

Rule 16. If the last syllable of a noun is hard and ends in non-sonorant consonants (b, v, g, d) or voiceless consonants (c, ç, f, k, q, p, s, ş, t, h), then the dative case ending 'ke' is added:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{b, v, g, d}, c, \text{ç}, f, k, q, p, s, \text{ş}, t, h\}}{\delta \cdot ke = \varphi,\ \ \varphi \in NDir} \tag{31}$$

Rule 17. If a noun ends in a back vowel, then an accusative case ending ‘nı’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \ \beta \in \{\mathrm{a}, \text{ä},\ \ \imath\ \}}{\delta \cdot n\imath = \varphi,\ \ \varphi \in NAcc} \tag{32}$$

Rule 18. If a noun ends in a front vowel, then the accusative case ending 'ni' is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \ \beta \in \{\mathrm{e, i}\}}{\delta \cdot ni = \varphi,\ \ \varphi \in NAcc} \tag{33}$$

Rule 19. If the last syllable of a noun is hard and ends in non-sonorant non-sonorants (j, z) or sonorant consonants (m, n, ň, y, w, r, l), then the accusative case ending ‘dı’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\mathrm{j, z, m, n, \text{ň}}, y, w, r, l\}}{\delta \cdot \mathrm{d\imath} = \varphi,\ \ \varphi \in NAcc} \tag{34}$$

Rule 20. If the last syllable of a noun is soft and ends in non-sonorant consonants (j, z) or sonorant consonants (m, n, ň, y, w, r, l), then the accusative case ending ‘di’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{j, z, m, n, \text{ň}}, y, w, r, l\}}{\delta \cdot \mathrm{di} = \varphi,\ \ \varphi \in NAcc} \tag{35}$$

Rule 21. If the last syllable of a is hard and ends in non-sonorant consonants (b, v, g) or voiceless consonants (c, ç, f, k, q, p, s, ş, t, h), then the accusative case ending 'tı' is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\mathrm{b, v, g, d, c, \text{ç}, f, k, q, p, s, \text{ş}, t, h}\ \}}{\delta \cdot t\imath = \varphi,\ \ \varphi \in NAcc} \tag{36}$$

Rule 22. If the last syllable of a noun is soft and ends in non-sonorant consonants (b, v, g) or voiceless consonants (c, ç, f, k, q, p, s, ş, t, h), then the accusative ending 'it' is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{b, v, g, d, c, \text{ç}, f, k, q, p, s, \text{ş}, t, h}\ \}}{\delta \cdot ti = \varphi,\ \ \varphi \in NAcc} \tag{37}$$

Rule 23. If a noun ends in a back vowel, then the locative case ending ‘da’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \beta \in \{\text{a}, \text{ä}, \text{ ı}\}}{\delta \cdot da = \varphi,\ \varphi \in Nloc} \tag{38}$$

Rule 24. If a noun ends in a front vowel, then the locative case ending ‘de’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \beta \in \{\text{e}, \text{i}\}}{\delta \cdot de = \varphi,\ \varphi \in Nloc} \tag{39}$$

Rule 25. If the last syllable of a noun is hard and ends in non-sonorant consonants (j, z) or sonorant consonants (m, n, ň, y, w, r, l), then the locative case ending ‘da’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\text{j}, \text{z}, \text{m}, \text{n}, \text{ň}, \text{y}, \text{w}, \text{r}, \text{l}\}}{\delta \cdot da = \varphi,\ \varphi \in Nloc} \tag{40}$$

Rule 26. If the last syllable of a noun is soft and ends in non-sonorant consonants (j, z) or a sonorant consonant (m, n, ň, y, w, r, l), then the locative case ending ‘de’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\text{j}, \text{z}, \text{m}, \text{n}, \text{ň}, \text{y}, \text{w}, \text{r}, \text{l}\}}{\delta \cdot de = \varphi,\ \varphi \in Nloc} \tag{41}$$

Rule 27. If the last syllable of a noun is hard and ends in non-sonorant consonants (b, v, g) or voiceless consonants (c, ç, f, k, q, p, s, ş, t, h), then the locative case ending 'ta' is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\text{b}, \text{v}, \text{g}, \text{c}, \text{ç}, \text{f}, \text{k}, \text{q}, \text{p}, \text{s}, \text{ş}, \text{t}, \text{h}\}}{\delta \cdot ta = \varphi,\ \varphi \in Nloc} \tag{42}$$

Rule 28. If the last syllable of a noun is soft and ends in non-sonorant consonants (b, v, g) or voiceless consonants (c, ç, f, k, q, p, s, ş, t, h), then p the locative case ending 'te' is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\text{b}, \text{v}, \text{g}, \text{c}, \text{ç}, \text{f}, \text{k}, \text{q}, \text{p}, \text{s}, \text{ş}, \text{t}, \text{h}\}}{\delta \cdot te = \varphi,\ \varphi \in Nloc} \tag{43}$$

Rule 29. If a noun ends in a back vowel, then the ablative case ending ‘dan’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \beta \in \{\text{a}, \text{ä}, \text{ ı }\}}{\delta \cdot dan = \varphi,\ \varphi \in NAbl} \tag{44}$$

Rule 30. If the noun ends in a front vowel, then the ablative case ending ‘den’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta,\ \beta \in \{\text{e}, \text{i}\}}{\delta \cdot den = \varphi,\ \varphi \in NAbl} \tag{45}$$

Rule 31. If the last syllable of a noun is hard and ends in non-sonorant consonants (j, z) or sonorant consonants (y, w, r, l), then the ablative case ending ‘dan’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{, \text{z}, \text{y}, \text{w}, \text{r}, \text{l}\}}{\delta \cdot dan = \varphi,\ \varphi \in NAbl} \tag{46}$$

Rule 32. If the last syllable of a noun is soft and ends in non-sonorant consonants (j, z) or sonorant consonants (y, w, r, l), then the ablative case ending ‘den’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{j, \mathrm{z}, \mathrm{y}, \mathrm{w}, \mathrm{r}, \mathrm{l}\}}{\delta \cdot \mathrm{den} = \varphi,\ \varphi \in NAbl} \tag{47}$$

Rule 33. If the last syllable of a noun is hard and ends in sonorant consonants (m, n, ň), then the ablative case ending ‘nan’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\mathrm{m}, \mathrm{n}, \mathrm{ň}\}}{\delta \cdot nan = \varphi,\ \varphi \in NAbl} \tag{48}$$

Rule 34. If the last syllable of the noun is soft and ends in sonorant consonants {m, n, ň}, then the ablative case ending ‘nen’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{m}, \mathrm{n}, \mathrm{ň}\}}{\delta \cdot nen = \varphi,\ \varphi \in NAbl} \tag{49}$$

Rule 35. If the last syllable of a noun is hard and ends in non-sonorant consonants (b, v, g, ğ, d) or voiceless consonants (c, ç, f, k, q, p, s, ş, t, h), then the ablative case ending ‘tan’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_h(\beta \cdot \gamma), \gamma \in \{\mathrm{b}, \mathrm{v}, \mathrm{g}, \mathrm{ğ}, \mathrm{d}, \mathrm{c}, \mathrm{ç}, \mathrm{f}, \mathrm{k}, \mathrm{q}, \mathrm{p}, \mathrm{s}, \mathrm{ş}, \mathrm{t}, \mathrm{h}\}}{\delta \cdot tan = \varphi,\ \varphi \in NAbl} \tag{50}$$

Rule 36. If the last syllable of a noun is soft and ends in non-sonorant consonants (b, v, g, ğ, d)) or voiceless consonants (c, ç, f, k, q, p, s, ş, t, h), then the ablative case ending ‘ten’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{b}, \mathrm{v}, \mathrm{g}, \mathrm{ğ}, \mathrm{d}, \mathrm{c}, \mathrm{ç}, \mathrm{f}, \mathrm{k}, \mathrm{q}, \mathrm{p}, \mathrm{s}, \mathrm{ş}, \mathrm{t}, \mathrm{h}\}}{\delta \cdot ten = \varphi,\ \varphi \in NAbl} \tag{51}$$

Rule 37. If a noun ends in a back and a front vowel, then the instrumental case ending ‘men’ is added to it:

$$\frac{\delta \in N,\ \delta = \alpha \cdot \beta,\ \beta \in V_1,\ \beta \in \{\mathrm{a}, \mathrm{ä},\ \mathrm{ı}\}}{\delta \cdot men = \varphi,\ \varphi \in NInst} \tag{52}$$

Rule 38. If the last syllable of a noun is soft and ends in sonorant consonants (m, n, y, y, w, r, l), then the instrumental case ending ‘men’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{m}, \mathrm{n}, \mathrm{ň}, \mathrm{y}, \mathrm{w}, \mathrm{r}, \mathrm{l}\}}{\delta \cdot men = \varphi,\ \varphi \in NInst} \tag{53}$$

Rule 39. If the last syllable of a noun is soft and ends in non-sonorant consonants (j, z), then the instrumental case ending ‘ben’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{j}, \mathrm{z}\}}{\delta \cdot ben = \varphi,\ \varphi \in NInst} \tag{54}$$

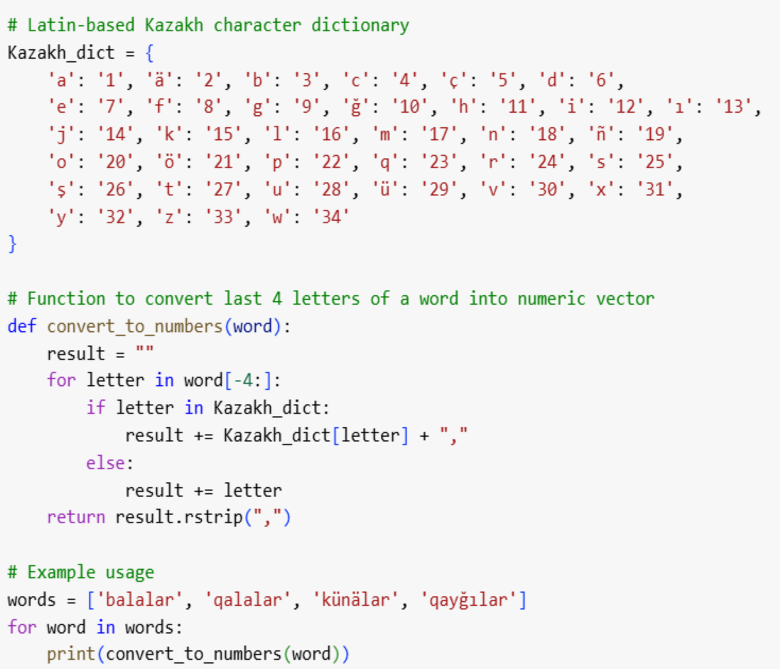
Rule 40. If the last syllable of a noun is soft and ends in voiceless consonants (c, ç, f, k, q, p, s, ş, t, h), then the instrumental ending ‘pen’ is added to it:

$$\frac{\delta \in N, \delta = \alpha \cdot \beta \cdot \gamma, S_s(\beta \cdot \gamma), \gamma \in \{\mathrm{c}, \mathrm{ç}, \mathrm{f}, \mathrm{k}, \mathrm{q}, \mathrm{p}, \mathrm{s}, \mathrm{ş}, \mathrm{t}, \mathrm{h}\}}{\delta \cdot pen = \varphi,\ \varphi \in NInst} \tag{55}$$

6. Representation of the Basic Patterns and Relations in the Nouns of the Kazakh Language Using Machine Learning

The purpose of this section is to analyze the created formal models of rules for constructing nouns of the Kazakh language in a new alphabet based on Latin graphics using neural networks. This model can be used to predict, evaluate or understand the behavior of the rules for the formation of nouns in the Kazakh language.

According to the new alphabet based on Latin graphics, we have formed a training dataset in JSON format of 1 000 words. Next, we wrote code to transform the last 4 characters into numbers in Python [22]. The function of the convert to numbers is to convert the characters into numbers according to the Latin-based Kazakh_dict dictionary. At the end of the algorithm, the conversion function starts, taking the meaning of every single word from word out of the prior prepared data set, designed in the form of a words dictionary. As a result, we get a set of {1,16,1,24}, {1,16,1,24}, {2,16,1,24}, {13,16,1,24}, {13,16,1,24} etc. We need these values to train a neural network. The Python code used to convert the last four characters of each Kazakh noun root into numerical vectors is shown in figure 2, along with the custom dictionary mapping each Latin letter to a unique integer ID.



```python
# Latin-based Kazakh character dictionary
Kazakh_dict = {
    'a': '1', 'ä': '2', 'b': '3', 'c': '4', 'ç': '5', 'd': '6',
    'e': '7', 'f': '8', 'g': '9', 'ğ': '10', 'h': '11', 'i': '12', 'ı': '13',
    'j': '14', 'k': '15', 'l': '16', 'm': '17', 'n': '18', 'ñ': '19',
    'o': '20', 'ö': '21', 'p': '22', 'q': '23', 'r': '24', 's': '25',
    'ş': '26', 't': '27', 'u': '28', 'ü': '29', 'v': '30', 'x': '31',
    'y': '32', 'z': '33', 'w': '34'
}

# Function to convert last 4 letters of a word into numeric vector
def convert_to_numbers(word):
    result = ""
    for letter in word[-4:]:
        if letter in Kazakh_dict:
            result += Kazakh_dict[letter] + ","
        else:
            result += letter
    return result.rstrip(",")

# Example usage
words = ['balalar', 'qalalar', 'künälar', 'qayğılar']
for word in words:
    print(convert_to_numbers(word))
```

Figure 2. Python implementation of the Kazakh Latin-based character vectorization algorithm for suffix modeling

A BR-BPNN with a layer size of 70 was constructed [23], [24], [25], [26]. The results of the training are presented in figure 1 and figure 3.

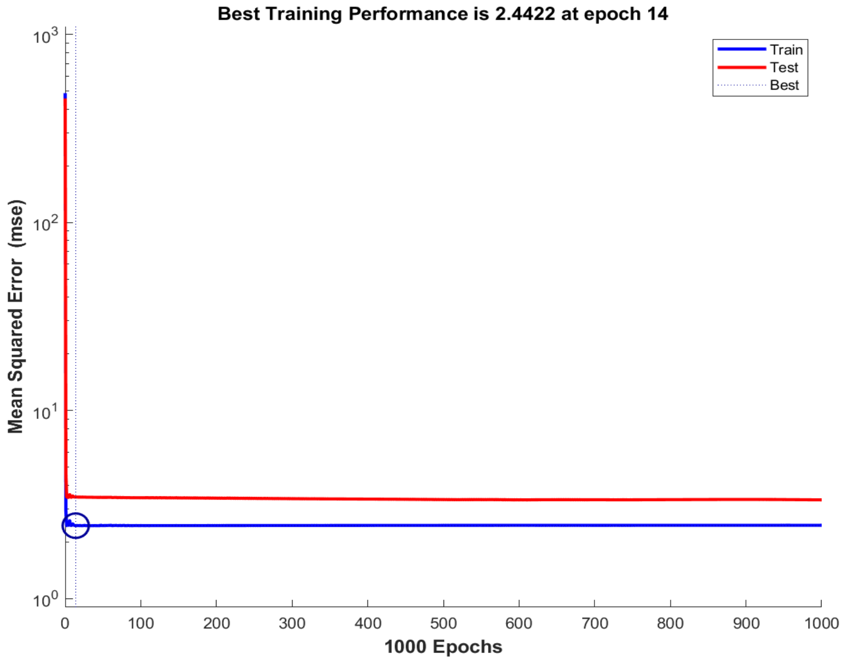


Figure 3. BR-BPNN Performance

BR-BPNN showed a low measure of the mean square of the difference between the predicted and actual values in the regression model. The training session showed the best performance at the 14th epoch and did not change much further. At the same time, the MSE was 2.4422. The measure of the strength and direction of the linear relationship between the variables R was 0.9876. When testing the neural model, R=0.98296, and the total R=0.98689.

Figure 4 and figure 5 shows the process of searching for a mathematical function that best matches a given set of data points. This is created with the help of regression analysis, this includes the search for function parameters that minimize the difference between the predicted values of the function and the actual meaning of the data points. The purpose of the fitting function is to create a model that can accurately represent the underlying patterns and relationships in the data. This model can then be used to predict, evaluate, or understand the behavior of the system being studied.

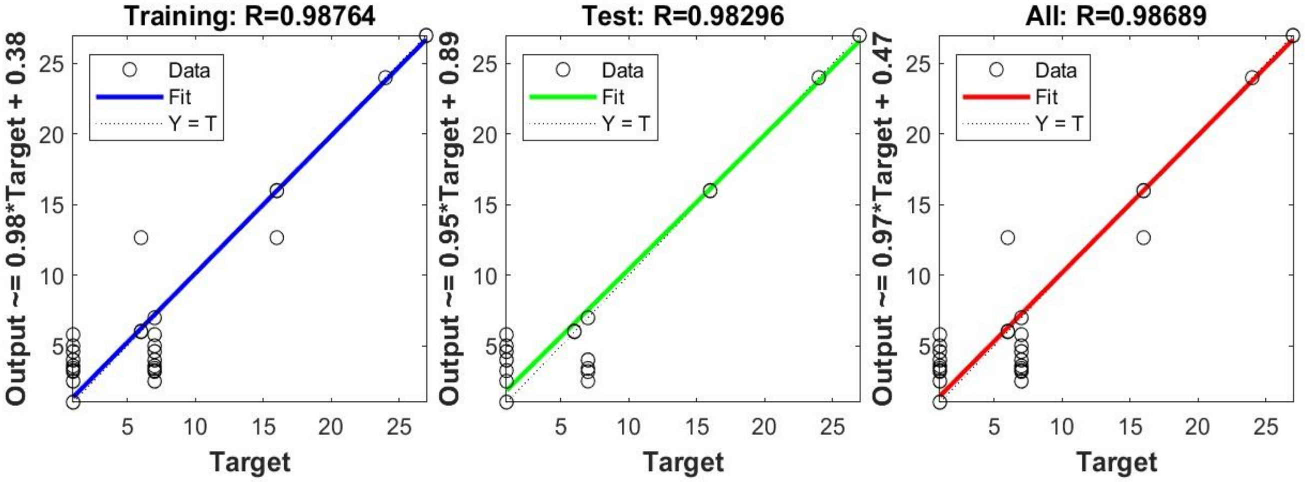


Figure 4. Displays the Correlation Level of the BR-BPNN Model

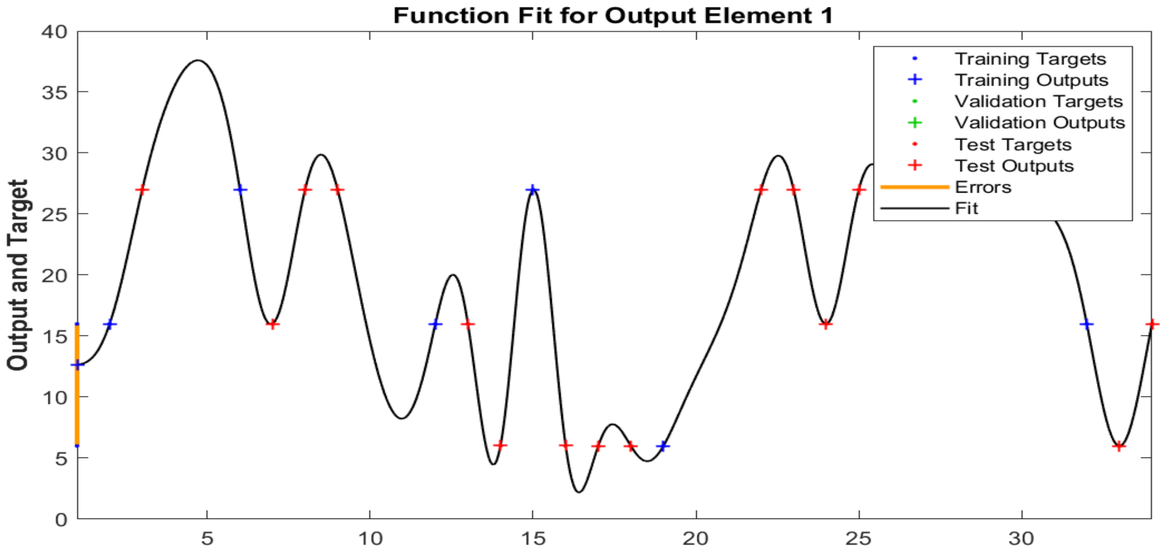


Figure 5. BR-BPNN Fit Function

7. Results and Error Analysis

The evaluation of the proposed system was conducted using a dataset of manually annotated noun forms represented in Latin script. The BR-BPNN model was trained on input–output pairs comprising base forms and their respective suffixes. Model performance was assessed using standard regression metrics MSE and the R. The achieved R-value exceeded 0.92, indicating a strong alignment between predicted and actual outputs, while the MSE remained low, demonstrating the model's capacity to effectively capture morphological patterns even with a relatively limited training dataset.

Although the results are promising, a detailed error analysis revealed specific cases where the model failed to generate the correct inflected forms. These errors were particularly common with root words ending in consonants that trigger complex vowel harmony rules. For instance, the model occasionally predicted suffixes that violated front–back vowel harmony, especially when the root contained mixed vowel environments. In such instances, although the regular-expression-based formal rules were applied correctly, the neural model produced incorrect suffixes due to insufficient representation of rare root structures in the training data.

Another category of errors arose in handling voiced–voiceless consonant alternations. Some plural forms required phonological alternation at the suffix boundary to preserve harmony (e.g., devoicing or voicing of final consonants), but the BR-BPNN model processed these as static sequences. This limitation indicates the absence of contextual phoneme-level features in the input encoding, which prevents the model from capturing subtle alternation rules integral to Kazakh morphophonology.

Furthermore, the model struggled with root forms that accept multiple valid suffix variants, depending on syntactic context or prosodic emphasis. These context-sensitive suffix options were not explicitly annotated in the training dataset, which introduces ambiguity and reduces learning precision. While the formal rule engine was capable of handling such cases through conditional expressions, the neural component lacked the necessary linguistic cues to distinguish among equally plausible suffixation outcomes. Table 5 presents a selection of suffix prediction results generated by the BR-BPNN model. It compares actual suffixes with predicted ones and provides an indicator of prediction accuracy. This comparative view highlights the model’s overall reliability while also illustrating typical error patterns in edge cases involving phonological complexity or contextual variation.

Table 5. Sample Prediction Results from BR-BPNN Model

Word (Latinized)	Actual Suffix	Predicted Suffix	Correct Prediction
Mektep (school)	ke	ke	Yes
Kitap (book)	qa	qa	Yes

aǵa (older brother/uncle)	men	ben	No
Qalam (pen/pencil)	men	men	Yes
ana (mother)	dan	dan	Yes

To address the identified limitations, several targeted improvements are proposed. First, the dataset should be expanded to incorporate a broader range of Kazakh noun roots, particularly those that exhibit phonological irregularities, voiced–voiceless alternations, or context-dependent suffixation. This expansion would enable the model to generalize more effectively across diverse morphological patterns.

Second, although the BR-BPNN model demonstrated reliable performance for the initial task, future iterations of the system could benefit from replacing or augmenting it with sequence-to-sequence (Seq2Seq) models or hybrid architectures that maintain BR-BPNN components while supporting character-level learning. Such models are better suited for morphologically rich and agglutinative languages, where sequence sensitivity and memory of prior inputs are essential. Finally, implementing feature engineering techniques, such as encoding vowel class, consonant type, or syllable structure, could enhance the model's ability to capture morphophonological rules and reduce errors associated with ambiguous or rare structures. Table 6 summarizes the most common error types identified during evaluation. For each error category, a brief description and corresponding improvement strategy are provided. This structured analysis serves as a guide for refining both the rule-based and neural components of the system in future development stages.

Table 6. Error Types and Description

Error Type	Description	Cause	Proposed Solution
Vowel harmony rule violation	Violation of front-back vowel harmony rules	The model failed on root words containing mixed vowel environments	Expand the dataset and encode vowel class features
Voiced-voiceless consonant alternation failure	Incorrect handling of voiced and voiceless consonant transitions	The model overlooked phonological alternations at suffix boundaries	Introduce contextual phoneme information into the input features
Context-sensitive suffix variation	Multiple valid suffix variants depending on syntax or emphasis	The dataset lacked explicit labels for context-sensitive variations	Include syntactic context or enhance the rule-based component with disambiguation rules

In conclusion, the integration of formal rules and neural modeling yielded a system that performs effectively on canonical forms while highlighting key limitations in handling linguistic exceptions. These findings provide valuable directions for future improvements in the development of robust Kazakh morphological analyzers. While BR-BPNN was utilized for initial suffix prediction experiments, the final BR-BPNN model further improved accuracy, as shown in Section 8.

8. Model Evaluation

To evaluate the effectiveness of the proposed BR-BPNN model for suffix prediction of Kazakh nouns in Latin script, we employed standard classification metrics. These include accuracy, precision, recall, and F1-score, which collectively provide a comprehensive assessment of the model's performance in capturing correct morphological transformations. The results are summarized in table 7, demonstrating the model's ability to generalize across diverse phonological patterns and suffixation rules with high reliability.

Table 7. Evaluation Metrics for the Proposed Model

Metric	Value (%)	Explanation
Accuracy	91.5	Proportion of total correct predictions
Precision	89.4	Proportion of predicted suffixes that were correct
Recall	90.2	Proportion of actual correct suffixes that were predicted
F1-score	89.8	Harmonic mean of precision and recall

The high-performance metrics indicate that the hybrid CNN-LSTM model is well-suited for capturing morphological patterns and predicting suffixes effectively in Kazakh noun forms.

9. Practical Applications

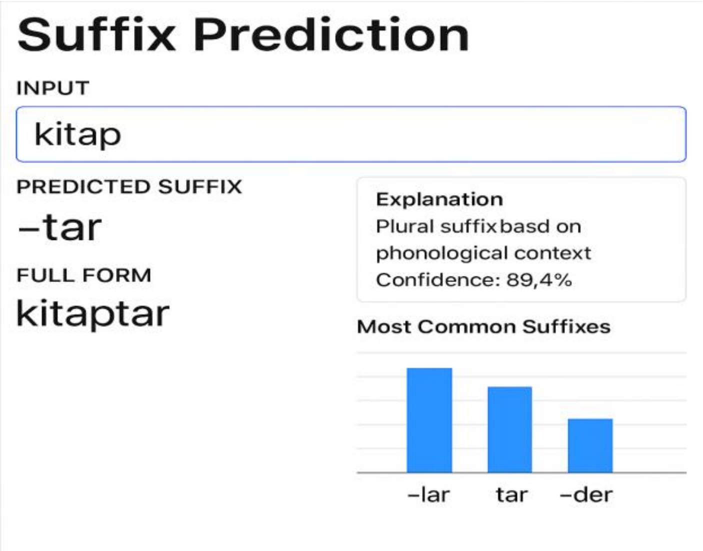
The models and methods proposed in this study are applicable in several domains of Kazakh language technology. First and foremost, they provide a foundational structure for developing spell-checkers, grammar correction tools, and intelligent text editors that are compatible with Latin-script Kazakh. These systems must accurately recognize and process various noun inflections, especially in morphologically complex cases involving case, number, and vowel harmony. In speech technologies, the formal modeling of noun morphology can improve the accuracy of Text-To-Speech (TTS) and Automatic Speech Recognition (ASR) systems. Since the endings of Kazakh words significantly impact pronunciation, particularly through vowel alternations and consonant softening, formal rules facilitate the prediction of correct phonetic forms during synthesis or recognition. This is particularly important for voice assistants and inclusive technologies intended for Kazakh-speaking users.

The models are also relevant for social media and political discourse analysis. As part of the AP19679847 project, the formal rules encoded in Latin script are being used to process and analyze user-generated texts related to elections, reforms, and civic activity. These texts are often produced by diaspora communities, which use inconsistent spelling or nonstandard grammar. The proposed approach offers a mechanism for normalizing these variations and extracting meaningful linguistic features for further semantic or sentiment analysis.

Additionally, this work contributes to the development of educational applications for teaching Kazakh grammar. Interactive exercises that generate or evaluate correct noun forms based on context can be implemented using the presented morphological rules. This is especially useful for learners accustomed to Latin script, including schoolchildren, second-language learners, or members of the diaspora. Finally, the proposed hybrid approach – combining rule-based formalism and machine learning – can serve as a framework for modeling other parts of speech or extending to additional Turkic languages. This adaptability opens possibilities for cross-linguistic NLP systems within the Turkic linguistic family.

9.1. Practical Interface of the Morphological Prediction System

To demonstrate the practical applicability of the proposed morphological prediction model, a conceptual interface of the system is presented in figure 6. This interface simulates a real-world usage scenario, whereby a user inputs a Kazakh noun written in the Latin script (e.g., kitap), and the system returns a predicted suffix (e.g., -tar), as well as the fully formed word (kitaptar).



Suffix Prediction

INPUT

kitap

PREDICTED SUFFIX

–tar

FULL FORM

kitaptar

Explanation
Plural suffix basd on
phonological context
Confidence: 89,4%

Most Common Suffixes

Figure 6. Example of a User Interface For Suffix Prediction Using the BR-BPNN Model for Latin-script Kazakh Nouns

The interface is designed to provide not only the prediction result but also accompanying interpretive elements that enhance user understanding. Specifically, it displays the confidence score of the prediction (e.g., 89.4%), along with a brief linguistic explanation that contextualizes the predicted suffix - for example, identifying it as a plural marker based on phonological features. In addition, a graphical visualization is presented to show the most frequently predicted suffixes for similar morphological patterns. Such an interface can be effectively incorporated into various practical applications, including digital platforms for teaching Kazakh as either a native or foreign language, automated spell-checking or grammar correction tools, and educational software designed to help users learn morphological rules and understand suffix usage in Kazakh texts written in Latin script. The development of this user-facing component highlights the potential of the proposed BR-BPNN-based model not only as a valuable research contribution but also as a foundational element in the creation of intelligent linguistic technologies tailored to Kazakh language processing.

10. Conclusion

This research has presented a formalized approach to modeling morphological rules for Kazakh nouns written in the new Latin alphabet. A hybrid architecture was employed, integrating formal linguistic rules with a BR-BPNN to learn and predict suffix formations based on structured phonological features. The construction of a representative dataset reflecting the phonotactic and morphosyntactic characteristics of the Kazakh language enabled the practical training and evaluation of the model.

Experimental results demonstrated that the proposed system achieves high predictive performance in generating morphologically valid noun forms, affirming the relevance of neural modeling for low-resource agglutinative languages. In addition to the algorithmic contributions, a conceptual interface was developed to visualize the prediction process, providing end-users with an interpretable output that includes suffix suggestions, confidence levels, and linguistic rationales.

The integration of such a system into language education tools, digital dictionaries, and intelligent writing assistants could significantly support the adoption and standardization of the Latin script in Kazakh, while also enhancing digital literacy and linguistic accessibility. Future research directions include expanding the morphological framework to encompass additional parts of speech, strengthening the system's linguistic explainability, and investigating cross-lingual transfer mechanisms for other Turkic languages.

11. Declarations

11.1. Author Contributions

Conceptualization: A.S., B.R., A.N.; Methodology: B.R.; Software: A.B.; Validation: L.Z., A.S., B.R.; Formal Analysis: L.Z., A.S., B.R.; Investigation: L.Z.; Resources: A.S.; Data Curation: B.R.; Writing – Original Draft Preparation: L.Z., A.S., B.R., G.B., A.N.; Writing – Review and Editing: A.B., A.N., B.Y.; Visualization: L.Z.; All authors have read and agreed to the published version of the manuscript.

11.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

11.3. Funding

This research is funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (grant no AP19679847).

11.4. Institutional Review Board Statement

Not applicable.

11.5. Informed Consent Statement

Not applicable.

11.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Turkic languages. [Online] [Accessed: Mar. 3, 2025].

[2] S. Altynbek, "The Kazakh Language Requires Reform of its Writing," *IgMin Research*, vol. 2, no. 2, pp. 073–083, 2024.

[3] L. Zhetkenbay, A. A. Sharipbay, G. T. Bekmanova, and G. K. Yelibaeva, "Method of the uniform morphological analysis of verbs Kazakh and Turkish languages," *Herald of the L.N. Gumilyov Eurasian National University*, vol. 6(121), no. 6, pp. 6–15, 2017. [in Kazakh].

[4] A. Sharipbayev, G. Bekmanova, B. Yergesh, and A. Mukanova, "Ontological models of morphological rules of Kazakh language in the form of semantic hypergraphs," *in Proceedings of the Open Semantic Technology for Intelligent Systems (OSTIS-2013), Minsk, Belarus*, 2013, vol. –, no. –, pp. 337–340.

[5] G. Bekmanova, A. Sharipbay, G. Altenbek, E. Adali, L. Zhetkenbay, U. Kamanur, A. Zulkhazhav, "A uniform morphological analyzer for the Kazakh and Turkish languages," *CEUR Workshop Proceedings*, vol. 1975, no. –, pp. 20–30, 2017.

[6] A.A. Sharipbay, B.Sh. Razakhova, A.S. Mukanova, and B.Zh. Yergesh, Mathematical and Ontological Models and Electronic Thesaurus of the Grammar of the Kazakh Language, Nur-Sultan, Kazakhstan: IP Bulatov A. Zh., 2020, p. 218.

[7] A. A. Sharipbaev, G. T. Bekmanova, A. K. Buribayeva, B. Z. Yergesh, A. S. Mukanova, A. K. Kaliyev, "Semantic neural network model of morphological rules of the agglutinative languages," *in Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and the 13th International Symposium on Advanced Intelligent Systems (ISIS), Kobe, Japan: IEEE*, vol. 2012, no. November, pp. 1094–1099, 2012.

[8] B. Yergesh, A. Mukanova, A. Sharipbay, G. Bekmanova, B. Razakhova, "Semantic hyper-graph-based representation of Noun in the Kazakh language," *Computacion y Sistemas*, vol. 18, no. 3, pp. 627–635, 2014.

[9] A. Sharipbay, B. Razakhova, A. Mukanova, B. Yergesh, G. Yelibayeva, "Syntax Parsing Model of Kazakh Simple Sentences," *in Lecture Notes in Computer Science*, vol. 10035, no. Dec., pp. 1–5, 2019.

[10] U. Kamanur, A. A. Sharipbay, G. Altenbek, G. Bekmanova, and L. Zhetkenbay, “Investigation and Use of Methods for Defining the Extents of Similarity of Kazakh Language Sentences,” *in Proc. 15th China National Conf. Chinese Computational Linguistics and 4th Int. Symp. NLP Based on Naturally Annotated Big Data (CCL–NLP-NABD), Yantai, China, Oct. 15–16, 2016, Lecture Notes in Computer Science*, vol. 10035, pp. 153–161, 2016.

[11] N. Zhumay, G. T. Zhiyembayeva, M. A. Zhunissova, J. A. Zhunissova, and S. Zhazira, "Lexemes with the 'camel' component in Kazakh: Problems of translation," *Opcion*, vol. 36, no. Special Ed. 27, pp. 1660–1674, 2020.

[12] L. Zhetkenbay, G. Bekmanova, B. Yergesh, and A. Sharipbay, "Method of Sentiment Preservation in the Kazakh-Turkish Machine Translation," *in Computational Science and Its Applications – ICCSA 2020, O. Gervasi et al., Eds., Lecture Notes in Computer Science*, vol. 12250, pp. 550–565, 2020.

[13] G. Kessikbayeva and I. Cicekli, "Rule-Based Morphological Analyzer of Kazakh Language," *Linguistics and Literature Studies*, vol. 4, no. 1, pp. 96–104, 2016.

[14] O. Makhambetov, A. Makazhanov, I. Sabyrgaliyev, and Z. Yessenbayev, "Data-Driven Morphological Analysis and Disambiguation for Kazakh," *in Proc. Int. Conf. Comput. Linguistics and Intelligent Text Processing (CICLing), Springer, Cham*, vol. 9041, pp. 151–163, 2015,

[15] A. Toleu, G. Tolegen, and A. Makazhanov, "Character-Aware Neural Morphological Disambiguation," *in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (Volume 2: Short Papers), Vancouver, Canada*, Jul. 2017, pp. 666–671,

[16] L. Zhetkenbay, A. Sharipbay, G. Bekmanova, and U. Kamanur, "Ontological modeling of morphological rules for the adjectives in Kazakh and Turkish languages," *Theoretical and Applied Information Technology*, vol. 91, no. 2, pp. 257–263, 2016.

[17] G. Yelibayeva, A. Mukanova, A. Sharipbay, A. Zulkhazhav, B. Yergesh, and G. Bekmanova, "Metalanguage and knowledgebase for Kazakh morphology," *in Proceedings of the 19th International Conference on Computational Science and Its Applications (ICCSA 2019), Saint Petersburg, Russia*, 2019, vol. 11619, no. no, pp. 693–706.

[18] S. Nakai, D. Beavan, E. Lawson, G. Leplâtre, J. M. Scobbie, J. Stuart-Smith, "Viewing speech in action: speech articulation videos in the public domain that demonstrate the sounds of the International Phonetic Alphabet (IPA)," *Innovation in Language Learning and Teaching*, vol. 12, no. 3, pp. 212–220, 2018.

[19] "Law of synchronism." [Online] [Accessed: Mar. 3, 2025].

[20] "Possessive endings." [Online] [Accessed: Mar. 3, 2025].

[21] E. Zhanpeisov, *Kazakh Grammar: Phonetics, Word Formation, Morphology, Syntax,* Astana, Kazakhstan: QT Publishing, 2002, p. 784. ISBN: 9965-16-345-6.

[22] A. Omarbekova, A. Sharipbay, A. Barlybaev, "Generation of test questions from RDF files using PYTHON and SPARQL," *in Proc. Conf. Control Engineering and Artificial Intelligence (CCEAI 2017), Kuala Lumpur, Malaysia*, vol. 806, no. 1, pp. 1–6, 2017.

[23] S. A. Abdymanapov, M. Muratbekov, S. Altynbek A. Barlybayev, "Fuzzy Expert System of Information Security Risk Assessment on the Example of Analysis Learning Management Systems," i*n IEEE Access*, vol. 9, pp. 156556-156565, 2021,

[24] A. S. Abdygalievich, A. Barlybayev and K. B. Amanzholovich, "Quality Evaluation Fuzzy Method of Automated Control Systems on the LMS Example," *in IEEE Access*, vol. 7, pp. 138000-138010, 2019,

[25] N. Amangeldy, M. Milosz, S. Kudubayeva, A. Kassymova, G. Kalakova, L. Zhetkenbay, A Real-Time Dynamic Gesture Variability Recognition Method Based on Convolutional Neural Networks. *Applied Sciences*, vol. 13, no. 19, pp. 10799, 2023.

[26] A. Barlybayev, L. Zhetkenbay, D. Karimov, B. Yergesh, "Development of a neuro-fuzzy model to predict the stocks of companies in the electric vehicle industry," *Eastern-European Journal of Enterprise Technologies*, vol. 4, no. 4(124), pp. 72–87, 2023.