3-бөлім **Раздел 3** **Section 3**

Информатика **Информатика** **Computer Science**

IRSTI 81.93.29 DOI: https://doi.org/10.26577/JMMCS2024-v123-i3-8

Sh.Zh. Mussiraliyeva, M.A. Bolatbek, Zh. Yeltay, K. Azanbay*
Al-Farabi Kazakh National University, Almaty, Kazakhstan
*e-mail: kuralayazanbay@gmail.com

DEVELOPMENT OF AN ERROR CORRECTION ALGORITHM FOR KAZAKH LANGUAGE

This article discusses a method for correcting spelling errors in the Kazakh language using the advantages of morphological analysis and a model based on noisy channels.
To achieve this goal, modern problems of automatic processing of Kazakh textual information were analyzed, existing linguistic resources and processing systems of the Kazakh language were systematized, the basic requirements for the development of a system for analyzing Kazakh textual information based on machine learning were determined, and models and algorithms for extracting facts from unstructured and poorly structured text arrays were developed.
The search function, an enhanced spelling correction algorithm, was utilized in this work and has the ability to recommend the proper spelling of the input text. The maximum editing distance, whether to include the original word when near matches are not found, and how to handle case sensitivity and exclusion based on regular expressions are all easily adjustable features of this functionality. Because of their adaptability, algorithms can be applied to a wide range of problems, from straightforward spell checks in user interfaces to intricate natural language processing assignments. Because of the way it's designed, the search function finds possible corrections and verifies the context of words while accounting for user preferences like verbosity and ignore markers.
Most modern multilingual natural language processing programs use only the graphical stage of text processing. On the other hand, semantic text analysis or analysis of the meaning of natural language is still an important problem in the theory of artificial intelligence and computational linguistics. But in order to process the grammar and semantics of multilingual information, pre-created semantic and grammatical corpora of each natural language are necessary. To solve this problem, several tasks were considered and solved. These tasks included the analysis of research in the field of machine learning methods used in the processing of textual information, the existing problems of formalization and modeling of the Kazakh language, as well as the development and implementation of models, methods and algorithms for morphological and semantic analysis of texts of the Kazakh language.

Key words: Kazakh language, dataset, monolingual datasets, leipzig corpora collection, Levenshtein distances, symspellpy, a multi-domain bilingual Kazakh dataset.

Ш.Ж. Мусиралиева, М.А. Болатбек, Ж. Елтай, Қ. Азанбай*
Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан
*e-mail: kuralayazanbay@gmail.com

Қазақ тілі үшін қателерді түзету алгоритмін әзірлеу

Бұл мақалада морфологиялық талдаудың және шулы арналарға негізделген модельдің артықшылықтарын пайдалана отырып, қазақ тіліндегі орфографиялық қателерді түзету әдісі қарастырылады.
Алға қойылған мақсатқа жету үшін Қазақ мәтіндік ақпаратын автоматты түрде өңдеудің қазіргі заманғы мәселелері талданды. Қазақ тілін өңдеудің қолданыстағы лингвистикалық ресурстары мен жүйелері жүйеленді және машиналық оқыту негізінде қазақ мәтіндік ақпаратын талдау жүйесін әзірлеуге қойылатын негізгі талаптар, сондай-ақ құрылымдалмаған ақпараттан фактілерді алу модельдері мен алгоритмдері айқындалды.

Бұл жұмыс іздеу функциясын, кіріс фразасының ықтимал дұрыс жазылуын ұсына алатын жетілдірілген емлені түзету алгоритмін қолданды. Іздеу функциясы оңай реттеледі және максималды өңдеу қашықтығының параметрлерін қолдайды және жақын сәйкестіктер болмаған кезде бастапқы терминді қосады. Сондай ақ регистрлерге негізделген регистр мен алып тастау сезімталдығын өңдейді. Бұл икемділік алгоритмдерді қарапайым UI емле тексерулерінен бастап табиғи тілді өңдеудің күрделі тапсырмаларына дейін әртүрлі жағдайларда тиімді пайдалануға мүмкіндік береді. Дизайнының арқасында іздеу функциясы ықтимал түзетулерді тиімді түрде анықтайды және ұсыныстарды елемеу маркерлері мен сөздік сияқты теңшелетін опцияларды ескере отырып, контекстке сәйкестігін тексереді. Қазіргі заманғы көп тілді табиғи тілді өңдеу бағдарламаларының көпшілігінде мәтінді өңдеудің графикалық кезеңі ғана қолданылады. Екінші жағынан, мәтінді семантикалық талдау немесе табиғи тілдің мағынасын талдау жасанды интеллект пен компьютерлік лингвистика теориясындағы маңызды мәселе болып қала береді. Бірақ көп тілді ақпараттың грамматикасы мен семантикасын өңдеу үшін әр табиғи тілдің алдын-ала жасалған семантикалық және грамматикалық корпустары қажет. Бұл мәселені шешу үшін бірнеше мәселелер қарастырылып, шешілді. Бұл міндеттерге мәтіндік ақпаратты өңдеуде қолданылатын машиналық оқыту әдістері саласындағы зерттеулерді талдау, қазақ тілін формализациялау мен модельдеудің қазіргі мәселелері, сондай-ақ қазақ тілі мәтіндерін морфологиялық және семантикалық талдау модельдерін, әдістері мен алгоритмдерін әзірлеу және іске асыру кірді.

Түйін сөздер: Қазақ тілі, деректер жинағы, бір тілді деректер жинағы, Лейпциг корпустарының жинағы, Левенштейн арақашықтығы, symspellpy, қазақ тіліндегі көп доменді екі тілді деректер жинағы.

Ш.Ж. Мусиралиева, М.А. Болатбек, Ж. Елтай, К. Азанбай*
Казахский национальный университет имени аль-Фараби, Алматы, Казахстан
*e-mail: kuralayazanbay@gmail.com

Разработка алгоритма исправления ошибок для казахского языка

В данной статье рассматривается метод исправления орфографических ошибок в казахском языке с использованием преимуществ морфологического анализа и модели, основанной на зашумленных каналах.

Для достижения поставленной цели были проанализированы современные проблемы автоматической обработки казахской текстовой информации, систематизированы существующие лингвистические ресурсы и системы обработки казахского языка, определены основные требования к разработке системы анализа казахской текстовой информации на основе машинного обучения, а также модели и алгоритмы извлечения фактов из неструктурированной информации. и были разработаны плохо структурированные текстовые массивы.

В этой работе использовалась функция поиска, усовершенствованный алгоритм коррекции орфографии, который мог предложить потенциально правильное написание входной фразы. Эта функция легко настраивается и поддерживает настройки максимального расстояния редактирования, включения исходного термина при отсутствии близких совпадений и обработки чувствительности к регистру и исключению на основе регулярных выражений. Такая гибкость позволяет алгоритмам эффективно использоваться в различных ситуациях, от простых проверок орфографии пользовательского интерфейса до сложных задач обработки естественного языка. Благодаря своей конструкции функция поиска эффективно выявляет потенциальные исправления и проверяет предложения на соответствие контексту, учитывая пользовательские параметры, такие как маркеры игнорирования и многословие. В большинстве современных многоязычных программ обработки естественного языка используется только графический этап обработки текста. С другой стороны, семантический анализ текста или анализ смысла естественного языка по-прежнему является важной проблемой в теории искусственного интеллекта и компьютерной лингвистики. Но для обработки грамматики и семантики многоязычной информации необходимы заранее созданные семантические и грамматические корпуса каждого естественного языка. Чтобы решить эту проблему, были рассмотрены и решены несколько задач. Эти задачи включали анализ исследований в

области методов машинного обучения, используемых при обработке текстовой информации, существующие проблемы формализации и моделирования казахского языка, а также разработку и реализацию моделей, методов и алгоритмов морфологического и семантического анализа текстов казахского языка. **Ключевые слова**: Казахский язык, набор данных, одноязычные наборы данных, коллекция корпусов leipzig, расстояния Левенштейна, symspellpy, многодоменный двуязычный набор данных на казахском языке.

1 Introduction

More than 12 million people speak Kazakh, which is the official language of the Republic of Kazakhstan. The Kazakh language is an interesting subject of research from the point of view of artificial intelligence, since it is an agglutinative language with complex morphology and relatively free word order [1].

Most of the existing multilingual natural language processing programs are limited to the grammatical stage of text processing. On the other hand, semantic text analysis or analysis of the meaning of natural language is still an important problem in the theory of artificial intelligence and computational linguistics.

Machine learning methods are becoming increasingly popular due to the fact that computer processors are becoming more powerful and rule-based methods require high intelligence costs.

But in order to process the grammar and semantics of multilingual information, precreated semantic and grammatical corpora of each natural language are necessary.

The main purpose of the work is to improve the quality of automatic text processing systems in the Kazakh language using machine learning methods and intelligent analysis models, as well as the development of models, methods and algorithms for analyzing the Kazakh text in order to determine the main characteristics of the text to create machine learning models.

For an input word with a typo (which may not be in the dictionary), find the nearest words from the dictionary based on a predefined metric. These found words are the correction options for the input word.

2 Literature review

Currently, various intelligent and mobile systems related to natural language processing are being actively created. Unfortunately, the issues of text processing of the Kazakh language are poorly developed, which hinders the development of information technology, and is associated with:

1) with the specifics of the Kazakh language as a language with a complex morphology;

2) with the lack of electronic resources for learning the Kazakh language in this area.

Nevertheless, the issues of text processing in Kazakh are very relevant in practice. An important problem is the problem of quickly finding specific words in documents. One of the ways to quickly search for words is to find the base of a word among the keywords of documents, allowing you to select the appropriate document as desired. One of the important processes in applied natural language processing systems, such as information retrieval, machine translation, etc., is normalization (lemmatization), i.e. bringing a word to its original basis[20].

Various scientists and scientific groups have carried out an analysis, and different approaches to the normalization of the Kazakh language have been considered. In the direction of segmentation of affixes of the Kazakh language, we can consider the work [20], where the morphemic structure in the corpus of the Kazakh language is analyzed, and the extraction of the basics and segmentation of the affix is studied. First, the finite-state machine (FSM) of inflectional affixes is installed, and then the segmentation of inflectional affixes is performed [20].

Our dataset was sourced from four domains: (1) digital mapping and navigation services (henceforth referred to as Mapping); (2) online marketplaces (henceforth referred to as Market); (3) an online library providing access to books and audiobooks in Kazakh (henceforth referred to as Bookstore); and (4) an online store offering a wide variety of applications for Android devices (hereafter Appstore). The dataset was gathered between September 2022 and September 2023, a duration of one year. Manual methods were used to gather reviews from Mapping and Market, while a BeautifulSoup script was used to gather feedback from Bookstore. The process of gathering Appstore evaluations was made easier by the utilization of the Python program google-play-scraper5.

A team of native Kazakh speakers manually reviewed each review. It was discovered that reviews frequently contained inappropriate content during the review process. However, no changes or deletions were made to this text in order to maintain the legitimacy and integrity of the reviews.

Consequently, Mapping provided 8,897 reviews that encompassed 407 institutions. Market contributed a significant amount of 30,289 reviews, covering 8,418 distinct products. Bookstore contributed 5,805 reviews, of which 3,792 were for audiobooks and 2,013 were for books. This means that there were 1,026 distinct audiobooks and books overall. Lastly, 135,073 distinct reviews of 1,759 Android games and apps were offered by Appstore. Of the users that contributed to these reviews, 31,490 users remained anonymous and 47,887 users had unique usernames.

Every review had a number score between 1 and 5, which allowed for a quantifiable depiction of people's opinions. As a result, in order to better represent its goal and substance, we titled the dataset KazSAnDRA /ka textprimstresssaendra/, an abbreviation for the Kazakh Sentiment Analysis Dataset of Reviews and Attitudes. A total of 180,064 reviews were gathered. The distribution of reviews across various scores and domains is shown in Table 1 [21].

Table 1. Domain and score statistics

Domain	Score					Total
	1	2	3	4	5	
Appstore	22,547	4,202	5,758	7,949	94,617	135,073
Bookstore	686	107	222	368	4,422	5,805
Mapping	959	270	369	525	6,774	8,897
Market	1,043	350	913	2,775	25,208	30,289
Total	25,235	4,929	7,262	11,617	131,021	180,064

There has been evidence of code-switching between Kazakh and Russian in Kazakhstan (Pavlenko, 2008), as well as a continuous transition from Cyrillic to Latin script. Thus, reviews that are considered to be in Kazakh may take various forms: among the possible

variations are: (a) only Kazakh words written in the Cyrillic script of Kazakhstan; (b) only Kazakh words written in Latin script; (c) a combination of Latin and Cyrillic characters; (d) a mixture of Russian and Kazakh words; or (e) a text written entirely in the Cyrillic script with Russian characters substituting Kazakh characters. Table 2 offers real reviews with examples of how they should be represented according to Kazakh spelling regulations and how to utilize the Cyrillic character for Kazakh words along with their English equivalent [21].

Table 2. Kazakh review variations

	Actual review	Correct form (Kazakh)	Correct form (English)
a	*керемет кітап*	*керемет кітап*	*a wonderful book*
b	*keremet*	*керемет*	*wonderful*
c	*jok кітап*	*кітап жоқ*	*no books*
d	*Осы приложениеге көп рахмет!*	*Осы қолданбаға көп рақмет!*	*Many thanks to this app!*

We used the dataset for two tasks to assess KazSAnDRA's efficacy: (a) polarity classification (PC), which entails determining whether a review is positive or negative, and (b) score classification (SC), which entails determining a review's score on a scale of 1 to 5. Reviews that began with a score of 1 or 2 in the PC task alone were flagged as unfavorable and given a new score of 0. Reviews that had previously received a score of 4 or 5 were, however, categorized as positive and given a new score of 1. Reviews that began with a score of three were deemed impartial and disqualified from the challenge. Regardless of the intended use of the information, the data pre-processing phase comprised multiple crucial procedures designed to maintain the consistency and integrity of the dataset. To start, every emoji was methodically taken out of the text in order to reduce any possible noise. All reviews were then lowercased for consistency and convenience of analysis. Punctuation was removed to make the text easier to read and absorb. To prevent interfering with later calculations, the characters for line breaks ($\backslash n$), tabs ($\backslash t$), and carriage returns ($\backslash r$) were also eliminated. A single space was consistently used in place of numerous spaces to improve readability and reduce needless mismatches [21].

3 Overview of existing data analysis methods

3.1 A set of text data

The Kazakh Language Corpus (KLC) [6] was assembled to help linguistics, computational linguistics and NLP research of the Kazakh language. It contains more than 135 million words in more than 400 thousand documents, classified by genre into the following five sections:

1) literary;
2) official;
3) scientific;
4) journalistic;
5) unofficial language.

KLC also has a piece of data with syntax and morphology annotations. It should be noted that initially the syntactic set of tags was a compact set of syntactic categories, which were

later improved during the development of the Dependency Tree of the Kazakh language 13, described below.

3.2 CC-100: Web scan data transformed into monolingual datasets

An attempt has been made to replicate the dataset used for XLR training with this corpus[7]. This corpus contains data for Romanized languages as well as monolingual data for over 100 languages. It was developed by processing Commoncrawl snapshots and utilizing URLs and paragraph indexes that were made available via the CC-Net repository [7]. Documents within documents are divided by double newline characters, while paragraphs within documents are separated by newlines. Each file contains these elements. The open source repository CC-Net is used to generate the data. Regarding the labor done in preparing the corpus, there are no rights to intellectual property.

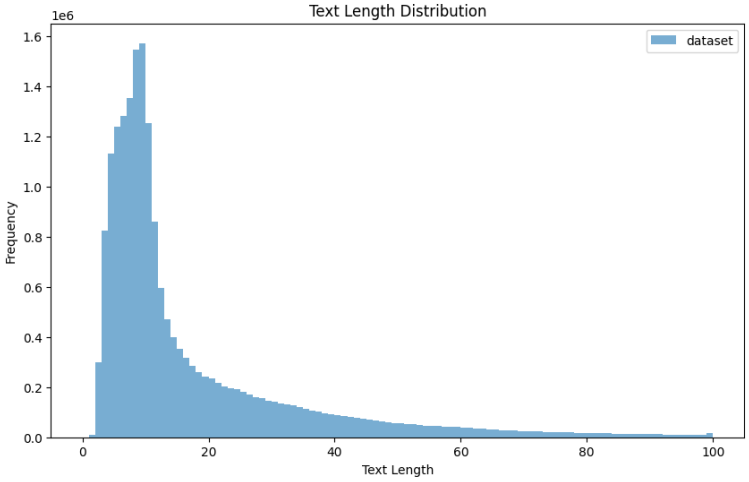

Figure 1: Distribution of the dataset

This effort aims at improving the interlanguage understanding of speech (XLU) through a detailed investigation of the consequences of large-scale acquisition of uncontrolled interlanguage representations. We introduce XLM-R [7], a multilingual masked language model built on a transformer that has been pre-trained on text in 100 different languages. It is equipped with the most advanced interlanguage categorization, sequencing, and question-answering capabilities available today [7].

Modern challenges of interlanguage understanding have been advanced by multilingual masked language models (MLM) like mBERT (Devlin et al., 2018) and XLM (Lample and Conneau, 2019) with joint pre-training of significant modifications of transformers (Vaswani et al., 2017) [8] in many languages.

As demonstrated by several benchmarks, such as interlanguage inferences in natural language (Bowman et al., 2015; Williams et al., 2017; Conn eau et al., 2018), question responses (Rajpurkar et al., 2016; Lewis et al., 2019), and named recognition (Pires et al., 2019; Wu and Dredze, 2019), these models effectively transmit across languages. All of these research, nevertheless, are carried out on Wikipedia, which has a very narrow scope, particularly for languages with fewer resources [8].

3.3 Leipzig Corpora Collection download page

You can download a variety of copyrighted tools and data from the Leipzig Corpora collection. The Leipzig Corpora collection uses comparable sources and the same style to deliver corpora in several languages. All information is supplied as text files, which may be used with the included import script to import the data into a MySQL database. They are meant to be used for applications like knowledge extraction systems as well as for scientific use by corpus linguists [9].

The cases are comparable in size, content, and presentation. They range in size from 10,000 to 1 million sentences and contain randomly picked sentences from the corpus language. Texts from newspapers or texts gathered at random from the internet serve as the sources. Sentences are used to separate the texts. Foreign-language offers and content have been removed. Additionally, word match information is pre-calculated and given because it is helpful for numerous applications. The most important words for each word are listed, whether they occur anywhere in the same phrase or as its immediate left or right neighbors.

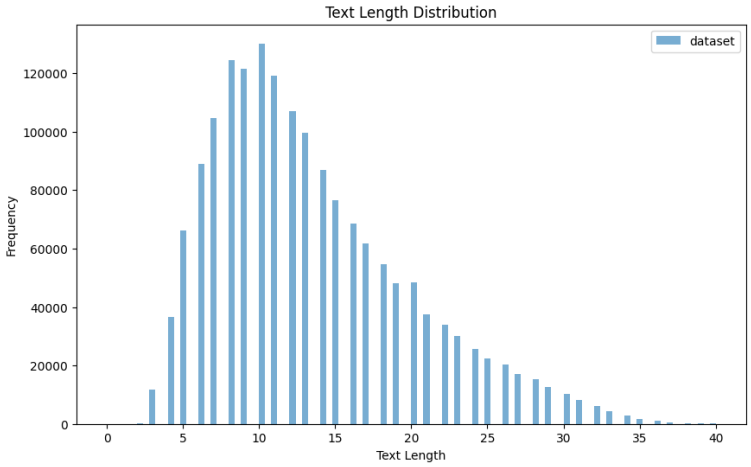

Figure 2: Text length distribution

The corpus is automatically assembled from carefully selected publicly available sources without detailed consideration of the content of the text contained. In particular, the views and opinions expressed in certain parts of the data remain solely with the authors [9].

4 Methods of using the dataset

The "Tree of Kazakh Dependence" is another significant linguistic resource that serves as the foundation for this work [13, 14]. A portion of the KLC was renamed using lexical, morphological, and syntactic annotations that computer scientists working on language processing issues can use, in accordance with guidelines [15] on universal Dependencies (UD) 23 for consistent grammar annotation. The same holds true for linguists. Approximately 61 thousand offers and 934.7 thousand tokens, of which 772.8 thousand are alphanumeric, are kept in the tree bank using the original UD CoNLL-U format. Additionally, labels including the individual's name, place, organization, and others were appended to each of the corps' proper names.

OSCAR 23.01

The January 2023 edition of OSCAR Corpus, known as OSCAR 23.01 [10], is based on the Common Crawl dump from November/December 2022. While it bears a lot of similarities to OSCAR 22.01, it comes with a few additional features: block list-based categories, precomputed locally sensitive hashes for near-deduplication, and adult content identification based on KenLM. Additionally, OSCAR 23.01 changed its compression method from gzip to standard [10].

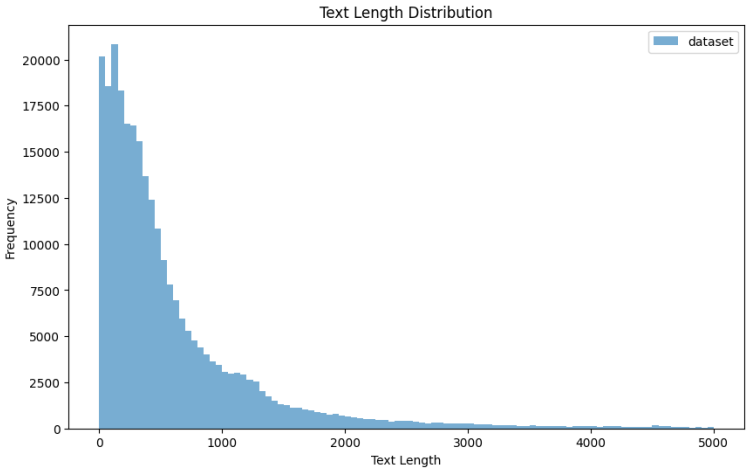

Figure 3: Using TLS to calculate the hash for each document

A hashing technique called location-based hashing generates comparable hashes for documents with similar contents. Both exact and nearly exact deduplication can be carried out with this. The hashes are the same for the same documents. As a result, all that is required to confirm identification is for papers that have the same hashes. A distance metric can be obtained by comparing TLS hashes. A threshold value of less than 40 yields a false positive level of 0.07% and a detection level of 49.6%, whereas a threshold value of less than 100 yields an FP level of 6.43% and a detection level of 94.5%, as stated in the original article [10].

Comparative analysis of the Levenshtein and Damerau-Levenshtein editorial distance algorithms

A popular variation of the Levenshtein distance that does not include the step of reordering adjacent letters is called the Damerau-Levenshtein distance. In other words, a weight of 1 is allocated rather than doing two delete and insert actions with a combined weight of 2. The Wagner-Fischer algorithm can be used to find the Levenshtein distance that is the shortest. The N-gram editorial distance simply employs the concept of Levenshtein distance as a symbol.

To find several copies of the text that is being considered in the document, the shingle method was developed. A single is a text segment made up of multiple words that have been processed for analysis. The shingle algorithm, often referred to as w-shingling, processes input data by employing a collection of shingles made up of N-grams, which are consecutive sequences of tokens in strings. The implementation consists of splitting strings into shingles, normalizing strings, checking for checksums, and looking for sequence matches.

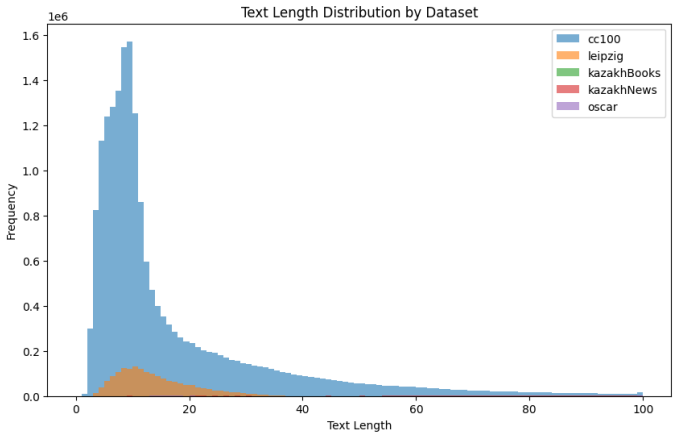

Figure 4: Comparative analysis of data set usage methods

When looking up search queries, typos-which happen when someone misses a key or inputs a query incorrectly-and spelling errors are the two most frequent kinds of mistakes. The most frequent user errors are displayed in Table 3.

Table 3. The most common mistakes

1	Replacement "и" – "е"	27
2	Replacement "а" – "о"	25
3	Extra space, the word must be written together	9.1
4	The absence of a space, instead of one word, two	8.5
5	Loss of one of the doubled letters	6.6
6	Replacing a deaf letter with a ringing letter and vice versa	3.6
7	Vowels after ц	2.7
8	Doubling a single letter	2.6
9	Loss "ь"	1.3
10	Excess "ь"	0.6
11	Replacement "ё" – "е"	0.1

Because automatic correction tools do not rely on sentence context, they can correct around 74% of typos. As a result, the accuracy of the chosen string comparison methods can be improved due to the prevalence of certain error types.

To determine how similar two strings that have been retrieved for comparison are, the shingle method is employed. The procedure is stopped if there are no differences; if not, the editorial distance is calculated and the next stage is initiated.

The fuzzy comparison of the received data gives rise to the fuzzy comparison of several rows problem. For fuzzy string comparison, a method based on figuring out the Damerau-Levenshtein distance values is applied. The Wagner-Fischer algorithm is used to calculate the separation between Levenstein and Damerau.

You can begin checking in accordance with the guidelines in table 1.1 if the editorial distance between two lines is one or less. Then, you can compare the outcomes with the information from the TAWT framework's morphological library [17]. In every other case, there's a good chance the strings will differ. The created algorithm's flowchart is displayed in Figure 5.

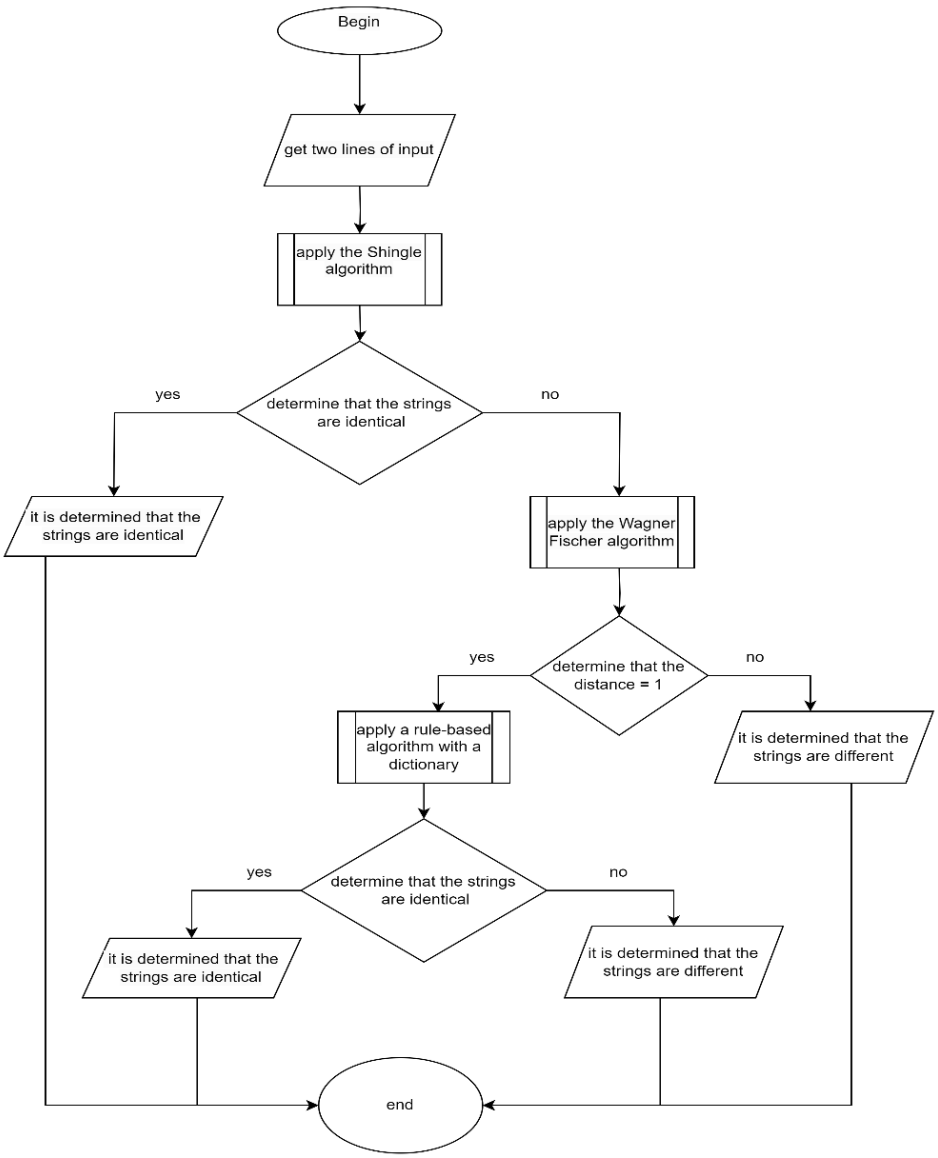

Figure 5: Block diagram of the typo search algorithm

Description of the symspellpy project

A Python port of SymSpell v6.7.1, known as Symspellpy, offers significantly faster performance and less memory usage. Port accuracy is ensured by implementing unit tests from the source project [12].

The Symmetric Delete spelling correction algorithm has made fuzzy search and spelling correction one million times faster.

For a particular Damerau-Levenshtein distance, the Symmetric Delete spelling correction algorithm simplifies the process of producing candidates for editing and dictionary search. This is six times faster than that [11].

In contrast to previous algorithms, just deletions-not transpositions, substitutions, or insertions-are needed. Dictionary term deletions result from transpositions, substitutions, and insertions of an input term. Language-specific substitutions and insertions are costly;

Chinese, for instance, has 70,000 Unicode Han characters!

The speed is attained by using low-cost functions to generate editing candidates that just require pre-calculation and deletion. With a maximum editing distance of three letters and an average of around 3 million spelling errors per five-letter word, SymSpell only needs to produce 25 deletions to cover every potential spelling error during both the initial computation and the search.

5 Results. The probability of words appearing

The Kazakh language dataset known as the multi-domain bilingual Kazakh dataset[18] has slightly more than 24,883,808 distinct texts across many domains. There are five divisions in total: kazakhBooks, leipzig, Oscar, cc100-monolingual-Crawled-data, and kazakhNews (Table 4). The dataset's statistics are displayed below:

Table 4. A multi-domain bilingual Kazakh dataset

Splitting the dataset (Dataset Split)	Domain	The number of texts in Split	Number of tokens in Split	The number of unique tokens in Split	The average number of tokens in the text
cc100-monolingual-crawled-data	Wikipedia articles	19 635 580	441 623 321	6 217 337	12
kazakhBooks	Books	8 423	351 433 586	7 245 720	40 264
leipzig	Articles/News	1 706 485	26 494 864	1 109 113	14
oscar	CommonCrawl	269 047	230 314 378	3 863 498	431
kazakhNews	News	3 264 273	1 041 698 037	5 820 543	209

In short, the idea is based on: words whose contexts are similar are most likely to have similar meanings. This implies how the typos in the first example are corrected. That is, we have two sentences: "адамның тилинде айту" and "адамның тілінде айту" , the contexts are similar, therefore, the words "тилинде"and "тілінде"are similar in meaning (this is a rough approximation, but the meaning is the same).

In addition to the obvious advantages, this approach to correcting typos has one important drawback – all the error variants that we can correct must be in the text we are learning from. That is, we can't get a vector for a word that we haven't seen before.

For example, the word "білім"(with the added characters of the beginning and end of the word, like ≪білім>") is converted into the following list: <былым, билим, блім, білм>. Then the resulting vector of the word is equal to the sum of the vectors of its n-grams:

$$V = \sum_{g \in G} z_g,$$

where,
G – the set of all n-grams of a word,
z_g – the vector of the corresponding n-gram,
V – the vector of the word.

It all helped to work in languages with rich morphology (such as Kazakh). Indeed, morphological changes now have less effect on the distance between words.

In the system of endings of the Kazakh language, all endings are divided into classes according to the length of characters. In a word, the ending of the maximum length for a given word is first searched for: it will be two characters shorter than the length of the word (it is assumed that the base cannot be less than length – 2). The expected end of length L is searched for in the corresponding class. If the ending is not in this class, then the length of the intended ending is reduced by one and searched in the corresponding ending class, etc., until an ending is found or the word is without an ending.

The Python code provided below defines the ‘edit 1(word)‘ function, which generates all possible corrections that are one correction away from the input word. Corrections include deletions, transpositions, substitutions, and insertions of characters.

This line combines all the fixes into a set to remove duplicates, and then returns this set of fixes.

```
def edits1(word):
    "All edits that are one edit away from `word`."
    letters    = 'abcdefghijklmnopqrstuvwxyz'
    splits     = [(word[:i], word[i:])    for i in range(len(word) + 1)]
    deletes    = [L + R[1:]               for L, R in splits if R]
    transposes = [L + R[1] + R[0] + R[2:] for L, R in splits if len(R)>1]
    replaces   = [L + c + R[1:]           for L, R in splits if R for c in
letters]
    inserts    = [L + c + R               for L, R in splits for c in
letters]
    return set(deletes + transposes + replaces + inserts)
```

As a result, the edit1 function uses the operations of deletion, transposition, replacement, and insertion to produce a set of words that are one correction apart from the input word. Lastly:

– Edits called deletions involve taking out a single character from a word.

– A transposition is an edit that involves swapping out two nearby characters.

– Substitutions involve editing a character by swapping it out for any letter in the alphabet.

– Inserts include moving a letter into any available space within a word.

Then, to guarantee uniqueness, the method returns a set that includes each of these modifications. Use this code to produce potential corrections for a given misspelled word in applications like spell checking or correction.

Table 5 provides a collection of sentences with misspelled words together with the corrected spellings of those terms.

Table 5. A selection of sentences with spelling errors

№	Original sentence	Spelling correction
1	араб тилин уйренем агылшынды уйренем дегениме дыл болды негизи бир еки жылдын колеми гой,	араб тілін үйренемін агылшынды үйренемін дегениме жыл болды негізі бир екі жылдың көлемі ғой
2	мен жумыс истегенде олар оздеримен жарысады бизде биреумен болса жапондықтар оздеримен казир сол адисти колданамын зато ешкимде шаруан жок,	мен жұмыс істегенде олар өздерімен жарасады бізде біреумен болса жапондықтар өздерімен қазір сол хадисте қолданамын зат ешкімге шаруа жок

3	бир кытай келип ана пракурордын бетине тукирсе урып жиберсе оган сот жок ал биздин алтын жигиттерге бары дау,	бир қытай келіп ана прокурордың бетіне тукирсе ұрып жіберсе оған сот жок ал біздің алтын жігіттерге бәрі жау
4	беттер құдайдан бергенде ой кудайдын жартысына шарасын занды бұзды деп тур деректер	менттер кудайдан безгендер гой кудайдын каргысына ушырасын занды бузды деп тур ешектер,
5	не қарап тұрғындар жағын айырсаңдарш	не карап турсындар жағын айырсаңдарш,
6	шіріген алмайтын иттер кредит керуен кетеді	шириген калмактын иттери уреди керуен кетеди

The following steps are used to correct the word "тилин" . The correction call(s) tries to select the most likely spelling correction for and. The need to find the appropriate letter (for example, correct "и" to "тілін" , or "талын" , or "тылын"), which involves the use of probabilities. In this paper, find an amendment among all possible corrections – "тілін" , which maximizes the probability that i is the intended correction, given the original word and:

argmaxc ∈ тілін P(и|i)

According to Bayes' theorem, this is equivalent to:

argmaxc ∈ тілін P(i) P(и|i) / P(и)

Since P(и) is the same for every possible tilin and, we can take this into account by giving:

argmaxc ∈ тилин P(i) P(и|i)

The four parts of this expression:

1. Selection mechanism: argmax

We choose "тілін" with the highest cumulative probability.

2. The "тілін, model: i ∈ "тілін" .

This tells us which possible c fixes to consider.

3. Language model: P(i)

The probability that c will appear as a word in the Kazakh text. For example, the appearance of "bir"is about 24% of the Kazakh text, so we should have P(бір) = 0.24 (Fig.)6).

4. Error model: P(и|i)

The probability that and will be printed in the text when the author meant i. For example, P(тилин|тілін) is relatively high, but P(талын|тілін) will be very low.

The Kazakh language is known as a multi-domain bilingual dataset, and contains more than 24,883,808 unique texts in many domains. The most common words in the data set in the Kazakh language are shown in Figure 6.

The search function is an advanced spelling correction algorithm designed specifically to suggest the potentially correct spelling of a given input phrase. This feature is easily configurable and supports setting the maximum editing distance, including the original term in the absence of close matches, as well as case sensitivity and exception handling based on regular expressions. This flexibility allows the algorithm to be effectively applied in a wide range of scenarios, from basic user interface spell checking to more complex natural language processing tasks.

In the operating mode, the function is started by matching the set maximum editing distance with the set limit, if exceeded, an error message is displayed. Depending on whether case transfer is enabled, it processes either the original or lowercase version of the input

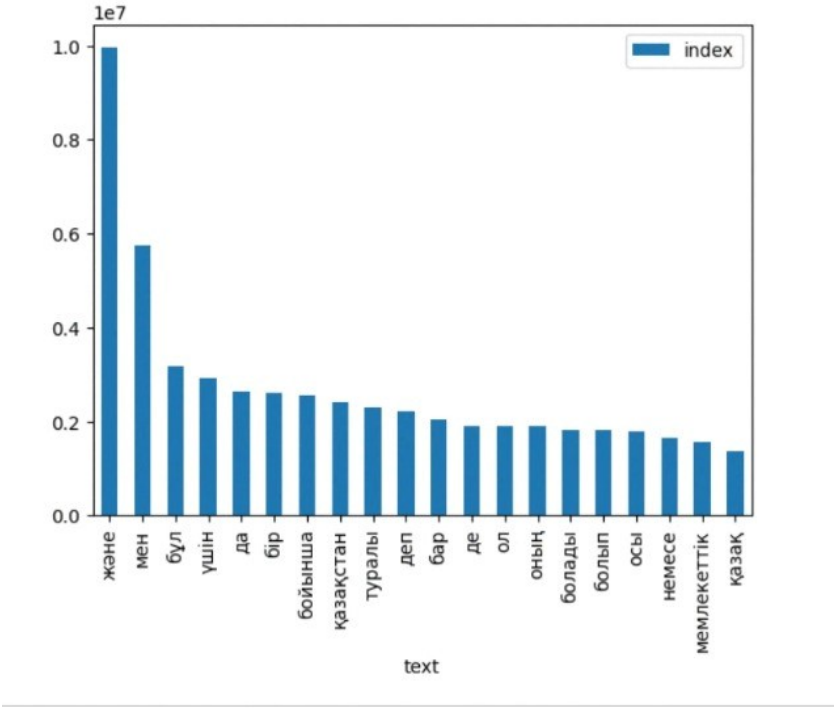

Figure 6: The probability of words occurring frequently in the dataset

data. The core of the algorithm includes an early exit strategy for rapid response in specific conditions and a careful search for exact matches in a predefined dictionary. If an exact match is not found, the algorithm generates possible corrections by a series of deletions, comparing each of them with the input phrase by calculating the Damerau-Levenshtein distance. This calculation takes into account insertions, deletions, substitutions, and transpositions to determine proximity to the original phrase.

The search function not only effectively identifies possible corrections, but also ensures that the sentences match the context, taking into account user-defined parameters such as verbosity and ignore markers. The received sentences are then sorted by editing distance and, if necessary, adjusted depending on the corpus, which makes this algorithm a reliable tool for improving the accuracy of text and user interaction in digital applications.

6 Conclusion

This article proposes a statistical and machine learning model of the Kazakh language. Experiments have demonstrated the model's effectiveness, yet it has the ability to move in different ways. This work will be enhanced in the future by investigating novel factors that influence object recognition. The dataset has been updated with new recommendations. Because our model works with a vast corpus, neural networks are incredibly effective at discovering named items in the data to create an outstanding model.

While using contemporary methods, such OSCAR 23.01, which combines CC-100 with neural networks, is straightforward and frequently yields positive outcomes, there are certain possible disadvantages. Words that were missed in the computation must be encoded as

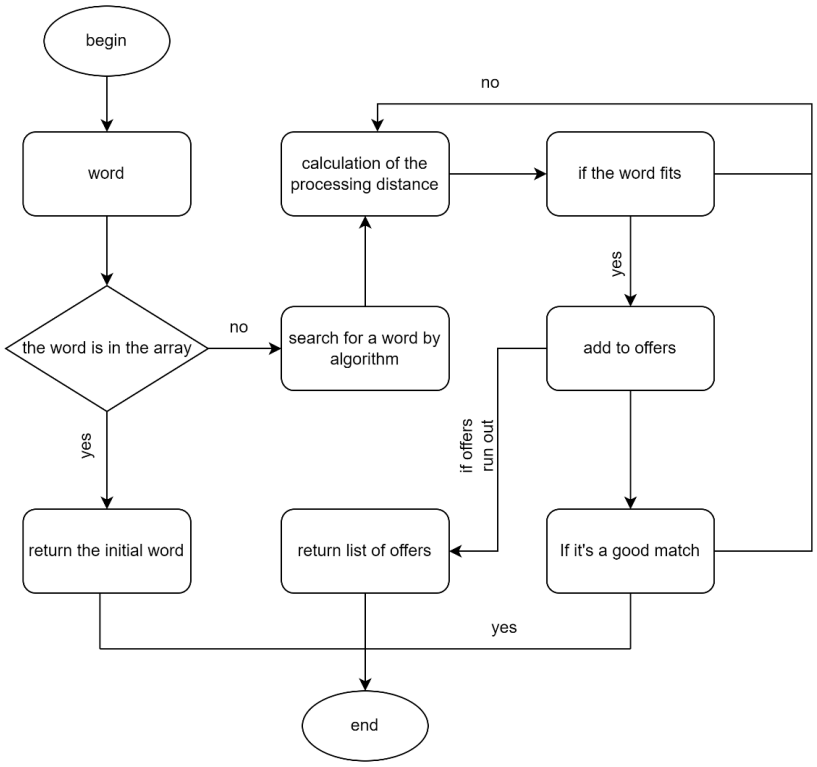

Figure 7: Spelling correction algorithm specially designed to suggest the possible correct spelling of a word

unknown and their meaning inferred from the words that surround them.

Fuzzy Search and Spelling Correction: The Symmetric Delete spelling correction method operates a million times more quickly. For a given Damerau-Levenshtein distance, the Symmetric Delete technique for spelling correction makes it easier to generate candidates for editing and dictionary searches.

The intricacies and contemporary issues with formalization in Kazakh semantics and grammar are examined. The features of Kazakh's automatic language processing in relation to its agglutinative language group membership are displayed. The language system's formalisms are taken into consideration, which need to be differentiated in order to develop a model for removing knowledge from its text and presenting it as a triplet of facts.

A semantic markup algorithm for Kazakh texts has been created. The fact triplets <Sub>, <Obj>, <Prec>, and the corpus's <POS type = "crime" tags define the semantic value of the token. The syntactic unit of a sentence or the grammatical information of a part of speech is determined by the value of the POS tags, and the criminal component of the token's semantic meaning is determined by the value of the type = "crime" attribute. The training dataset for the Kazakh language is just a collection of suffixes and linguistic norms in simple markup.

Acknowledgement

This research was carried out within the framework of the project "Multiclassification of ideological trends of cyber extremism in the Kazakh language using artificial intelligence

methods", funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (grant No AP19676342, project manager Mussiraliyeva Sh.).

References

[1] Yessenbayev Zh., Kozhirbayev Zh., Makazhanov A., *KazNLP: A Pipeline for Automated Processing of Texts Written in Kazakh Language* (2020) DOI:10.1007/978-3-030-60276-5_63.

[2] Cheng V., Li Ch., "Combining Supervised and Semi-supervised Classifier for Personalized Spam Filtering. In: Zhou, ZH., Li, H., Yang, Q. (eds) Advances in Knowledge Discovery and Data Mining. Lecture Notes in Computer Science(), vol 4426", *Springer, Berlin, Heidelberg* (2007). https://doi.org/10.1007/978-3-540-71701-0_45.

[3] Kessikbayeva G., Ilyas Cicekli. "Rule Based Morphological Analyzer of Kazakh Language", *In Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM,* Baltimore, Maryland. Association for Computational Linguistics 46–54. DOI: 10.3115/v1/W14-2806.

[4] Assylbekov Zh., Washington J., Tyers F., Nurkas A., Sundetova A., Karibayeva A., Abduali B., Amirova D. "A free/open-source hybrid morphological disambiguation tool for Kazakh", (2016).

[5] Washington J., Salimzyanov I., Tyers F. "Finite-state morphological transducers for three Kypchak languages", *In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14),* Reykjavik, Iceland. European Language Resources Association (ELRA). (2014): 3378–3385.

[6] Makhambetov O., Makazhanov A., Yessenbayev Zh., Matkarimov B., Sabyrgaliyev I., Sharafudinov A. "Assembling the Kazakh Language Corpus", *In Proceedings of the Conference on Empirical Methods in Natural Language Processing,* Seattle, Washington, USA. Association for Computational Linguistics (2013): 1022–1031.

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, Veselin Stoyanov, "Unsupervised Cross-lingual Representation Learning at Scale", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL),* (2020): 8440–8451.

[8] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, Edouard Grave, "CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data", *Proceedings of the 12th Language Resources and Evaluation Conference (LREC),* (2020): 4003–4012.

[9] Goldhahn D., Eckart T., Quasthoff U., "Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages", *In: Proceedings of the 8th International Language Resources and Evaluation (LREC'12),* (2012).

[10] https://oscar-project.github.io/documentation/versions/oscar-2301/

[11] https://pypi.org/project/symspellpy/#modal-close

[12] https://github.com/wolfgarbe/SymSpell

[13] Makazhanov A., Sultangazina A., Makhambetov O., Yessenbayev Z., "Syntactic annotation of Kazakh: Following the universal dependencies guidelines", *A report. In proceedings of the 3rd International Conference on Turkic Languages Processing,* Kazan, Tatarstan (20153): 338–350.

[14] Makazhanov A., Yessenbayev Z., "NLA-NU Kazakh Dependency Treebank", https://github.com/nlacslab/kazdet

[15] Nivre, J.; https://universaldependencies.org/

[16] Makazhanov A., Makhambetov O., Sabyrgaliyev I., Yessenbayev Zh., Spelling Correction for Kazakh, 2014.

[17] Фреймворк TAWT [Электронный ресурс]: - Режим доступа: https://textanalysis.ru/jce/details/tawt

[18] https://huggingface.co/datasets/kz-transformers/multidomain-kazakh-dataset#data-splits

[19] https://norvig.com/spell-correct.html

[20] Rakhimova D.R., Turganbaeva A.O., "Normalization of Kazakh language words", *Scientific and Technical Journal of Information Technologies, Mechanics and Optics,* 20 (4) (2020): 545–551 (in Russian). Doi: 10.17586/2226-1494-2020-20-4-545-551.

[21] Yeshpanov Rustem, Huseyin Atakan Varol, "KazSAnDRA: Kazakh Sentiment Analysis Dataset of Reviews and Attitudes" , *Astana, Kazakhstan,* (2024).

Information about authors:

Mussiraliyeva Shynar – Candidate of Physical and Mathematical Sciences, Docent, Department of Information Systems of the Al-Farabi Kazakh National University (Almaty, Kazakhstan, email: mussiraliyevash@gmail.com);

Bolatbek Milana – Ph.D., Senior Lecturer, Department of Information systems of the Al-Farabi Kazakh National University (Almaty, Kazakhstan, email: bolatbek.milana@gmail.com);

Azanbay Kuralay (corresponding author) – Master of science, teacher, Department of Information systems of the Al-Farabi Kazakh National University (Almaty, Kazakhstan, email: kuralayazanbay@gmail.com);

Yeltay Zhastay – Master of science, Al-Farabi Kazakh National University (Almaty, Kazakhstan, email: jastayeltay@gmail.com);

Авторлар туралы мәлімет:

Мүсіралиева Шынар – физика-математика ғылымдарының кандидаты, доцент, әл-Фараби атындағы ҚазҰУ ақпараттық жүйелер кафедрасы (Алматы қ., Қазақстан, email: mussiraliyevash@gmail.com);

Болатбек Милана – Ph.D., әл-Фараби атындағы ҚазҰУ ақпараттық жүйелер кафедрасының аға оқытушысы (Алматы қ., Қазақстан, email: bolatbek.milana@gmail.com);

Азанбай Құралай (корреспондент автор) – ғылым магистрі, әл-Фараби атындағы ҚазҰУ ақпараттық жүйелер кафедрасының оқытушысы (Алматы қ., Қазақстан, email: kuralayazanbay@gmail.com);

Елтай Жастай – ғылым магистрі, әл-Фараби атындағы ҚазҰУ (Алматы қ., Қазақстан, email: jastayeltay@gmail.com);

Received: May 19, 2024
Accepted: June 20, 2024