The presented paper is devoted to the development of a method for identifying and correcting spelling errors in Kazakh texts. In this paper, the study object is methods for more accurate correction of spelling errors in Kazakh texts. The aim of the study is to develop an augmented version of the Damerau-Levenshtein method for correcting spelling errors in Kazakh language texts. Automatic detection and correction of spelling errors have become a default feature in modern text editors for working with text data, in text messaging applications such as chatbots, messengers, etc. However, although this task is well solved in geographically widespread languages, it has not been fully solved in languages with a small audience, such as the Kazakh language. The methods developed so far cannot correct all spelling errors found in Kazakh texts. Therefore, the development of a method with specific algorithms for spelling error correction in Kazakh texts is considered. As a result of the research work, algorithms for correcting errors found in Kazakh language texts were developed, and the developed algorithms were included in the Damerau-Levenshtein method. The experimental testing results of the augmented Damerau- Levenshtein method showed 97.2 % accuracy in correcting specific errors found only in Kazakh words and 92.8 % accuracy in correcting common errors from letter symbols. The standard Damerau-Levenshtein method testing results showed 76.4 % accuracy in correcting specific errors found only in Kazakh words. The results of the tests in correcting common errors from letter symbols with the standard Damerau-Levenshtein were approximately the same with the augmented Damerau-Levenshtein method, the accuracy is 92.2 %. The extent and conditions of practical application of the results are implemented by including them in text editors, messengers, e-mails and similar applications that work with text data

Keywords: NLP, algorithm, text data, probability, spelling error, edit distance, similarity

UDC 004.912
DOI: 10.15587/1729-4061.2023.289187

DEVELOPMENT OF AN AUGMENTED DAMERAU–LEVENSHTEIN METHOD FOR CORRECTING SPELLING ERRORS IN KAZAKH TEXTS

Nurzhan Mukazhanov
PhD, Associate Professor*
Zhibek Alibiyeva
PhD, Associate Professor*
Aigerim Yerimbetova
Corresponding author
PhD, Associate Professor, Leading Researcher
Institute of Information and Computational Technologies
Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan
Shevchenko str., 28, Almaty, Republic of Kazakhstan, 050010
Professor*
E-mail: a.yerimbetova@satbayev.university
Aizhan Kassymova
PhD, Deputy Director
Institute of Automation and Information Technologies**
Nursulu Alibiyeva
Senior Teacher
Al-Farabi Kazakh National University
Al-Farabi ave., 71, Almaty, Kazakhstan, 050040
*Department of Software Engineering**
**Satbayev University
Satpayev str., 22a, Almaty, Kazakhstan, 050013

Received date 13.08.2023
Accepted date 17.10.2023
Published date 30.10.2023

***How to Cite**: Mukazhanov, N., Alibiyeva, Z., Yerimbetova, A., Kassymova, A., Alibiyeva, N. (2023). Development of an augmented damerau–levenshtein method for correcting spelling errors in kazakh texts. Eastern-European Journal of Enterprise Technologies, 5 (2 (125)), 23–33. doi: https://doi.org/10.15587/1729-4061.2023.289187*

1. Introduction

Spelling error correction is an important and desirable feature for modern applications that use textual information. For example, e-mail, chatbots, instant messengers, text search engines, text editors, etc. It is one of the most important research tasks in natural language processing (NLP). In many cases, NLP tasks, applications and programs require spelling error correction at the first stage of textual data processing.

A lot of research has been done on the development of spelling checker methods, and many open-source and commercial applications have been developed. The first research work on spell correction was published in the 1960s [1] and continues to this day [2, 3]. Methods and areas of application for correcting errors in texts directly depend on their environment and the characteristics of natural languages [4]. Errors encountered in textual data during processing natural languages can be divided into two types. It can be said that errors of the first type or common errors are similar in all natural languages. They are factors related to textual errors in several languages, such as missing letters in words, accidentally pressing other letters, not knowing the correct spelling of a word, etc. Errors of the second type or specific errors are errors due to the specific features of each natural language. Accordingly, different methods of correcting such errors are required [5].

The developed methods and applications are not able to solve all spelling error correction problems in Kazakh texts [6]. Because, as a natural language, the Kazakh language has its own peculiarities in word formation and specific differences in spelling errors. Modern search engines, text

editors, messengers and software systems do not provide the ability to correct Kazakh language text errors. According to the performed test results on the search engines, such as google.com and yandex.com, we noticed that these systems use spelling error correction techniques of English and Russian languages for processing other natural languages. Consequently, functions such as processing search queries, correcting errors, and suggesting correct options do not work correctly in natural languages with a small audience.

Therefore, studies on the development of specific methods taking into account natural language features are relevant.

2. Literature review and problem statement

Many research works were carried out on spelling error correction in widely used international languages, such as English [1], Russian [7], Chinese [8], Arabic [9], Indian [10], etc., and diverse methods were developed. In the paper [1], a method of spelling error correction based on insertion, deletion, and replacement operations was proposed. Although this work was a prerequisite for future research and development of spelling error correction methods, it does not cover all spelling rules and concerned only the English language. In [7], the authors studied the method of error correction of Russian-language texts obtained from the Internet. The task of text error correction was solved by a combination of Levenshtein's distance algorithm, dictionary search and hypothesis generation using machine learning. In the paper, the focus is on Russian-language words that are changed by users, and from the point of view of orthography, they are translated into incorrectly written words. The Levenshtein method uses the insertion, deletion, and replacement operations to construct correct word versions. Since this method does not use the operation of transposition (changing the position of adjacent letters), the exact correct word may be missed. Since 1 out of 4 correction operations are not used, the chance of identifying the correct word immediately drops by 25 percent. In addition, the word formation of the Russian language is completely different from the Kazakh language. The work [8] is devoted to the correction of semantic and grammatical errors in Chinese texts. However, in the paper, the Parts-of-speech matching rules are given in English. If we consider that Chinese is a different language from English, a way to overcome these difficulties in the implementation of Chinese text error detection and repair framework based on an online learning community can be impossible.

In [9], the authors studied the problem of correcting space deletion errors in Arabic texts. The practical implementation of the research results is not given in the work. Despite this, the authors provided information on the correction rate. The Levenshtein's method was also used in text error correction. A legitimate question arises as to why the Damerau-Levenshtein method, an improved form of this method, was not considered.

The paper [10] presents the results of research on spelling error correction for the Indian language. The work provides systematic information on ways to correct text errors and tools like Bangla Spell Checker, Marathi Spell Checker, Malayalam Spell Checker, and more. However, the practical implementation of the research results is not considered. In addition, the features of Kazakh language text error correction are not taken into account in the considered text error correction plugins.

Research has been done on spelling error correction in languages such as Amazigh [11], which have a small audience, but there are not many such works. From the review of studies on developing a spelling error correction method for different natural languages, it was determined that the text error correction method is directly related to the word formation, spelling rules and specific features of the natural language. Therefore, a spelling correction method developed for one language should not be considered completely suitable for another language.

Depending on the scale, there are also studies related to specific applications. The paper [12] presents the results of research using spell correction for the information retrieval task. In this paper, the distance method of text error correction is used to correct the wrong word in the search query by inserting, deleting, replacing and inserting one space. As the task is focused on only information retrieval and is intended for English, it does not satisfy our problem solving. The paper [13] is devoted to the study of spelling error correction for search engines. In addition to improving the accuracy of spelling error correction, the authors also set the task of optimizing the search engine. But there were fully unresolved issues related to spelling error correction, because it is limited to correcting a single letter error. There is a reason for this, if correcting an error for more than two letters, the algorithm becomes more complicated and affects search engine optimization.

In addition to individual languages, there have also been studies of corrections adapted to several natural languages depending on the text content [14]. The work mentions the ability to correct spelling errors that can be adapted to 24 languages. This task is carried out in systems with high computing power, because the rules of each language must be taken into account in the algorithms. The complexity of the algorithm, which takes into account the peculiarities of each language, increases. In addition, using the vocabulary of each natural language requires additional memory. These complications indicate the difficulty of using the proposed solution in small mobile applications, messengers, etc.

So far, it can be said that very few scientific papers have been written on spell correction during typing texts in Kazakh [6, 15]. The main difference between the research conducted in the papers [6] and [15] is that in [15] the authors considered the correction of spelling errors by means of a morphological disambiguator, and in [6] the authors investigated using finite state automata for spell correction of Kazakh synthetic texts. In the work [6], it is proposed to use morphological analysis and create rules for the Kazakh language using a finite automaton. However, describing all the rules of natural language is a very difficult task and occurs in word formations that do not obey strict rules. The reason for this may be difficulties in practical implementation. In [15], the correct version of a misspelled word is formed by morphological analysis. Probability calculation is used to determine the correct version. Morphological analysis does not give all possible correct versions of separate words because it depends on the grammatical meaning of the word in the sentence, and error checking of a single word has difficulties in giving correct results. Nevertheless, this method works better when considering the whole sentence.

In [16], the authors reviewed many methods for correcting spelling errors. As a result of the review, the most

frequently used methods and solutions are identified. The results of studies and solutions on spelling error correction suggested that the most used solution is the mixed solution. In this case, the methods are used in combination and complement each other. These decisions were of a recommendatory nature in the development of our algorithm. An algorithm for correcting a specific spelling error was not developed and presented in the work, so it was not of practical importance for us.

In [17], the authors used the Jaro-Winkler method as a word similarity search method. Using this method, the similarity of words can be determined. In the presented work, only one symbol distance checking is considered, but the main emphasis is on reducing the search time.

In [18], a rule-based spelling and grammatical error correction method is provided. Applying a rule-based approach to spelling error correction requires describing all natural language rules in an algorithm. For languages with rich vocabulary and complex word formation, this is a very difficult task and may not be feasible in practice, which makes relevant research impractical.

We noticed from the review of research works that their approaches were used for solving the first type of spelling errors. However, to solve specific error problems of a certain natural language, an addition to existing methods or a new solution is required. There were unresolved issues related to specific errors in Kazakh texts. All this suggests that it is advisable to conduct a study on the development of a special method for spelling error correction for each natural language, in particular for the Kazakh language.

3. The aim and objectives of the study

The aim of the study is to develop an augmented version of the Damerau-Levenshtein method for correcting spelling errors in Kazakh texts.

To achieve this aim, the following objectives are accomplished:

– to create a model for correcting identified spelling errors and include all types of error correction structures in the model;

– to develop an algorithm for solving specific spelling error problems in Kazakh texts;

– to carry out experimental testing of the developed algorithms.

4. Materials and methods of research

4. 1. Object and hypothesis of the study

The object of the research is methods for more accurate correction of spelling errors in Kazakh texts.

The main hypothesis of the study assumes that the effectiveness and accuracy of spelling error correction methods depend on the features of the natural language, the types and specifics of errors found in texts. The solution of spelling error correction tasks can be divided into two subtasks: the first is to identify misspelled words, and the second is to correct errors.

The following assumptions were made:

– identifying wrong words correctly is the first step in solving the task. The computer determines whether words are correct or incorrect using logical expressions, predefined rules, special searches, etc. based on the given structures. It is very important to clearly define the correctness of the word;

– to identify the types of spelling errors found in texts. Searching for solutions based on error types allows developing a more accurate solution. The problem can be solved by developing a completely new method or in an innovative way by improving existing methods.

In order to evaluate the performance of the proposed method, experimental testing was carried out to compare its accuracy and efficiency with other modern error correction methods.

4. 2. Error identification methods

As a result of the review of scientific papers, it becomes clear that the most used methods for detecting errors are *N*-gram [19] and Dictionary Look-up analysis [11].

N-gram analysis. In general, an *n*-gram is considered to be a sequence of *n*-elements [20]. *N*-gram has a wide range of applications, including mathematics, biology, geology, etc. It is used in solving text data processing problems, solving problems such as clustering texts or words, determining word sequences, determining letter sequences, creating bigrams. One of the tasks solved by this method is the identification of misspelled words in the text. An *n*-dimensional square matrix is created, consisting of *n*-gram frequencies. If a missing or rare *n*-gram is found in the checked word, the word is marked as a misspelled one. In the algorithm, each string involved in the comparison process is divided into sets of adjacent *n*-grams. The similarity between two strings is achieved by finding the number of unique *n*-grams that they share, and then calculating the similarity coefficient, that is, the number of common *n*-grams (intersection) divided by the total number of *n*-grams in the string [4].

Vocabulary search. A dictionary is a base (corpus) consisting of the correct words of a particular language. Each word in the text is checked for presence in the dictionary. If the word is present, it is considered spelled correctly, if not, then it is considered as a misspelled word. In order to correctly find misspelled words, the dictionary must contain all the words of the language and be constantly updated with new words. The basic forms of the word are stored in some dictionaries designed to detect the misspelled word, and the base of the word being checked is determined by morphological analysis, and it is checked whether it is correct or wrong [14].

In this work, we use the vocabulary search method to identify misspelled words. The vocabulary search method is often used in natural languages that have a formed corpus and show exactly whether a word is correct or incorrect. In order to determine the misspelled word, a database of Kazakh words was created, and the correctness or misspelling of the word is checked by searching. And in *N*-gram analysis, many operations are required to form *n*-grams, and *n*-grams do not always return the correct answer. But this method is language-independent as it does not require knowledge of the language being used.

4. 3. Spelling error correction methods

Many methods are used to correct textual errors. Among them, the following methods are often used: Hamming distance, Levenshtein distance, Damerau-Levenshtein distance, Jaro distance, Rule-Based, *N*-gram, Probabilistic, etc. [19].

Hamming distance is the number of different characters in the same positions in two strings of the same length

(sequences of characters or words). This method measures the minimum number of replacements required to replace one string with another, or the minimum number of errors that can be converted from one string to another [19]. This method helps to determine the difference between two words or strings of the same length. However, if there are extra or missing characters in the word, then it is impossible to correct the error.

Levenshtein distance is a method of measuring the similarity/difference between two sequences of characters (words). The value of the method is the smallest (minimum) number of conversion operations required to replace or convert one word (w_m) to another word (w_c). Modification operations include addition, deletion, and replacement. When converting one string to another, they do not have to be the same length. The Damerau-Levenshtein distance method was developed by complementing the transformation operations in the Levenshtein distance method [20].

Damerau-Levenshtein distance is a method of measuring the similarity/difference between two words (or strings). The Damerau-Levenshtein distance is defined as the minimum number of operations of insertion, deletion, substitution, and transposition of two adjacent characters required to change or transform one word (w_m) into another one (w_c). As a result of substitutions, a set of candidate words $\{w^1_{cand}, w^2_{cand}, w^3_{cand}, ..., w^k_{cand}\}$ is obtained from the misspelled word w_m. The distance between the candidate word and the misspelled word is determined by the function $d_{w_m,w_c}(i,j)$ (2):

$$d_{w_m,w_c}(i,j) = \begin{cases} 0 \text{ if } i=j=0, \\ d_{w_m,w_c}(i-1,j)+1 \text{ if } i>0, \\ d_{w_m,w_c}(i,j-1)+1 \text{ if } j>0, \\ d_{w_m,w_c}(i-1,j-1)+1_{(w_{m_i}\neq w_{c_j})} \text{ if } i,j>0, \\ d_{w_m,w_c}(i-2,j-2)+1_{(w_{m_i}\neq w_{c_j})} \\ \text{if } i,j>1 \text{ and } w_{m_i}\neq w_{c_{j-1}} \text{ and } w_{m_{i-1}}\neq w_{c_j}, \end{cases} \quad (2)$$

where, i – misspelled word (w_m) characters,

j – candidate word (w_c) characters. $1_{(w_{m_i}\neq w_{c_j})}$ – exponential function, if $w_{m_i}=w_{c_j}$ the exponential function is 0, otherwise the exponential function is 1.

This algorithm is one of the most commonly used text error correction algorithms, various software developments have been developed with experimental tests. For example, in the research paper [21], linear space algorithms were developed to calculate the Damerau-Levenshtein (DL) distance between two strings and determine the optimal number of correction operations. In addition, experiments were carried out and a faster version of the algorithm was presented. To implement the Damerau-Levenshtein distance algorithm, dynamic programming is used [4, 16, 21].

Jaro distance – a method for measuring the distance between two strings (w_{mis} and w_c). Based on the Jaro method, the Jaro-Winkler similarity detection method was developed. This method first finds the Jaro distance and then sets the scale factor p. The scale factor is recommended to be given equal to 0.1, in some cases other values are given, but it should not exceed 0.25. The Jaro-Winkler method improves similarity accuracy by providing a scale factor [17].

Rule-Based – in this method, a set of rules is formed on the basis of many grammatical requirements of natural language and regular combinations of words in it in the form of an *n*-gram. Verification of words is carried out by comparison with the *n*-gram. If an incorrect *n*-gram is found in a word, it is replaced by a correct *n*-gram according to a given rule. This method does not require storing all the correct words of the language. In addition to word errors, it is widely used to detect and correct grammatical errors in sentences [18, 22].

Probabilistic methods – are based on some statistical features of the language. Detection and correction of textual errors are based on the Bayes rule, using the creation of *n*-gram methods developed in language models. In the probabilistic correction of text errors, two methods are widely used – transition or Markov probabilities and confusion probabilities. Transition probabilities depend on the language, and confusion probabilities depend on the source (language corpus) [10, 23].

Specific errors in Kazakh texts. To correct the first type of errors, the Damerau-Levenshtein method has been used in many studies. However, to correct specific errors in Kazakh texts, a special structure and algorithm should be developed. There are specific errors in Kazakh texts as follows:

– using "alternative" letters of the Russian alphabet {а, и, к, г, у, у, о, х} instead of the letters of the Kazakh alphabet {ә, і, қ, ғ, ү, ұ, ө, һ}. For example: "болашаққа қадам" is typed as "болашакка кадам". Today, in messengers, e-mails, and social networks, there are many incorrectly written texts using alternative letters of the Russian alphabet instead of Kazakh letters. Even when filling out official documents, there are those who make such mistakes;

– since the Kazakh language is used in Kazakhstan along with the Russian language, when typing without changing the keyboard from Russian to Kazakh, instead of Kazakh letters {ә, і, қ, ғ, ү, ұ, ө, һ}, numbers and characters {2, 3, 0, 5, 8, 9, -, =} are written (the keyboard layout can be seen in [26]). For example, "болашаққа қадам" is typed as "болаша00а 0адам";

– typing without changing the keyboard layout from Latin to Kazakh. For example, "болашаққа қадам" instead of "jkfis00f 0flfv". At the same time, typing words in the Kazakh alphabet in Latin letters, for example, "болашаққа қадам" instead of "bolashaqqa qadam". This problem was solved in search engines like google.com, yandex.com and it is not essential to repeat existing solutions.

Keyboard in Kazakh. Today, the Cyrillic alphabet is officially used for the written Kazakh language. The alphabet consists of 42 letters. In it, 33 letters { А, Б, В, Г, Д, Е, Ё, Ж, З, И, Й, К, Л, М, Н, О, П, Р, С, Т, У, Ф, Х, Ц, Ч, Ш, Щ, Ъ, Ы, Ь, Э, Ю, Я } are taken from the Russian alphabet, and 9 letters {Ә, Ғ, Қ, Ң, Ө, Ұ, Ү, Һ, І} correspond to the phonetic features of the Kazakh language [24, 25]. The layout of letters on the keyboard, made according to this alphabet is shown in [26]. Specific letters of the Kazakh language are placed in the top row of the keyboard.

The first of the above errors is very common in Kazakh texts. Many users have gotten into the habit of writing with alternate letters and reached the point that they don't care if it is wrong. The next, second, and third mistakes are not common, but they do happen. To make text applications more user-friendly and to save time, auto-correction functionality is needed. Kazakh is the state language in Kazakhstan and 20 million people on earth use the Kazakh language. Therefore, it has a 20 million audience in digital spaces, it also makes sense to develop NLP algorithms for the Kazakh language.

5. Results of research on the development of a spelling error correction method for Kazakh texts

5. 1. Spelling error correction model

The structure of the model for correcting spelling errors in Kazakh words is shown in Fig. 1. The blocks of the proposed model are similar to the models for correcting text errors in English and other languages, but the internal algorithms in the text processing blocks are based on the alphabetic spelling of the Kazakh language.

To correct errors, any Kazakh text can be taken. Initial processing is carried out according to the received text. Texts can be of any length and typed on different devices: $Initial\ text = \{w_{it_1}, w_{it_2}, w_{it_3}, \ldots, w_{it_p}\}$. Next, misspelled words are detected, $misspelled\_words = \{w_{m_1}, w_{m_2}, w_{m_3}, \ldots w_{m_j}\}$. Dictionary lookup is used to identify misspelled words. To create the dictionary, the frequency dictionary of the Kazakh language was used, and the frequency of occurrence of words in the dictionary was stored as a key-value. By creating a dictionary of unique words, the misspelled word is accurately determined. $D_u = \{w_1, w_2, w_3, \ldots, w_i, \ldots, w_n\}$ is a dictionary of unique words. In addition, frequencies of words are included in the dictionary $D_u = \{\{w_1 : v_{w_1}\}, \{w_2 : v_{w_2}\}, \{w_3 : v_{w_3}\}, \ldots, \{w_i : v_{w_i}\}, \ldots, \{w_n : v_{w_n}\}\}$, we also called it the corpus of Kazakh words. One by one, the words of the source text are checked for presence or absence in the dictionary. If a non-existent word is found in the dictionary, the word is added to the list of misspelled words.

In the third block, words are generated that are at a n – distance from the misspelled word. All generated words $gen\_words = \{w_{gen_1}, w_{gen_2}, w_{gen_3}, \ldots, w_{gen_l}\}$. A modified form of the Damerau-Levenshtein distance algorithm has been developed to determine the distance between misspelled words in the Kazakh text and the correct variant. In this block, replace to Kazakh letters and replace numbers to letters algorithms were added to the Damerau-Levenshtein method to generate the correct version of misspelled Kazakh words. In the proposed model, the correction of misspelled words is performed first using the replace to Kazakh letters and replace numbers to letters algorithms.

In the fourth block, the list of candidate words $cand\_words = \{w_{c_1}, w_{c_2}, w_{c_3}, \ldots, w_{c_e}\}$ is filtered from the generated words of the 3D block. If there are several candidate words for one misspelled word, the probability of words is calculated in the fifth block.

In the sixth block, variants of the expected correct variants of the misspelled word are proposed. If the correction algorithms produce only one correct version, that version is proposed as the proper correction. If there are several similar words, then the word with the highest probability is predicted as the correct variant.

It was found that 80 percent of typing errors are caused by a single letter [1]. In texts in the Kazakh language, errors in typing one letter instead of another, skipping a letter, overwriting one letter, changing the position of neighboring letters are the same as in other languages [9]. These errors can be called typical errors, and they can be solved using the insert, delete, swip, and replace operations in the Damerau-Levenshtein distance method. Each operation has its own algorithm. The use of this method when correcting errors in different languages can be seen in the following scientific papers [4, 21].

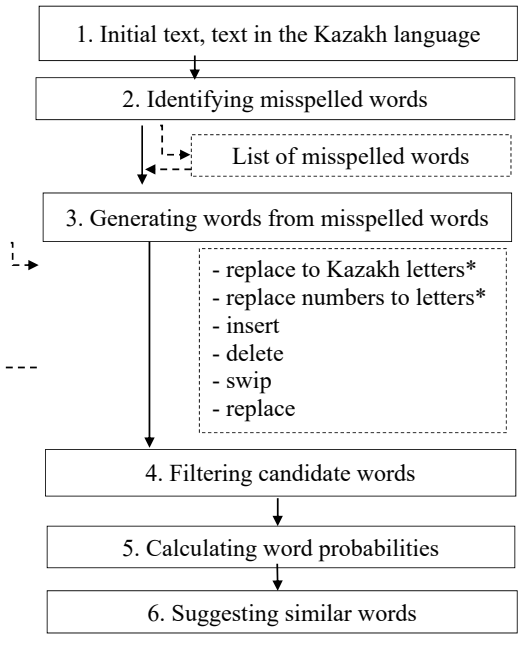

Fig. 1. Model for correcting errors in Kazakh texts

In addition to typical mistakes, the first of the most common mistakes in Kazakh texts is the use of "alternative" letters of the Russian alphabet instead of Kazakh letters (Fig. 2). The main reason is that not all users have the Kazakh language keyboard enabled, mainly on mobile devices. Because of this, in most cases, the Kazakh language text is typed in alternative Russian letters, for example, the word "сәрсенбі күні" is typed as "сарсенби куни".

Another type of error is typing without changing the keyboard from Russian to Kazakh on computers and laptops. In this case, instead of Kazakh letters { ә, і, қ, ғ, ү, ұ, ө, һ }, the corresponding numbers and symbols from the keyboard are written {2, 3, 0, 5, 8, 9, –, =}. These errors are not as common as the previous ones, but occur when typing the source text in text editors, when writing queries to search engines, and are corrected later. For example, the word "сәрсенбі күні" is typed as "с2рсенб3 к8н3", which is difficult to read and understand.

Words can be corrected by using Russian letters instead of Kazakh letters or using the Damerau-Levenshtein substitution algorithm for wrong numbers and characters "–, =". But, if we use this algorithm, too many correction operations are performed and too many words are generated. The complexity of the replacement algorithm (replace) for one letter in a word is $O(M*N)$ operations and requires the same amount of memory. In some words written in Russian letters, it will be necessary to replace several letters. Then the complexity of the algorithm will be very large. If numbers are typed instead of Kazakh letters, then the numbers "2, 3, 0, 5, 8, 9" and the signs "–, =" should be added to the sequence of characters.

This means that if we add 6 digits and 2 digits to the 42 letters of the Kazakh language, then for any replacement we will need to use 50 characters.

If we take the number of characters used in the substitution as M=50 and, for example, if we use the substitution to correct the word "с2рсенб3" (Wednesday in English). To replace one character $O(M1)=M*N=400$ (N is the word length), a replacement operation is performed. In our case, it is necessary to replace two characters, it will be necessary

to use combinations that replace one character. Then the complexity of replacing two characters is $O((M*N)M*N)$= =160,000. There are words in which the characters {ә, і, қ, ғ, ү, ұ, ө, h} occur four or five times. When correcting such words, the replace operation makes a huge combination. As a result, the computational complexity is high and the combinatorial words to be generated are large. When replacing the generated characters by generation, completely different words may appear. If the constructed words are present in our vocabulary and the Bayesian probability calculation gives a high value, we will have several candidate words. This will prevent you from giving the correct version of the misspelled word. If we replace the alternative letters of the Russian language with the corresponding Kazakh letters, take only words with numbers and symbols "–, =" and replace only these numbers and symbols from them, then the complexity of the algorithm will decrease.

Replace to Kazakh letters. Correction of Kazakh texts typed in Russian letters is an urgent task in text editors, chatbots, mobile devices, and primary processing of search queries. In erroneous texts, instead of the letters of the Kazakh alphabet {ә, і, қ, ғ, ү, ұ, ө, h}, letters of the Russian alphabet are used { а, и, к, г, у, о, х } (Fig. 2).

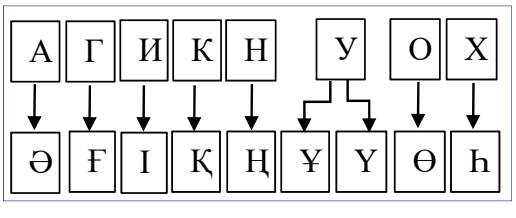

Fig. 2. Russian letters instead of Kazakh ones

The first task of correcting words typed with letters of the Russian alphabet is to determine the number of potentially replaceable letters in a word. Using the "combination" (combinations) formula of combinatorics, it is possible to determine the number of all variants generated by individual combinations of variants (all variants of the character sequence formed by the combination).

For example, if we consider the word "САРСЕНБИ", then 3 letters can be replaced by "А, Н, И" (Fig. 3). If we count the number of combinations from 1 to k=3, we will end up with 7 combinations. Therefore, the number of generated versions is seven {СӘРСЕНБИ, САРСЕҢБИ, САРСЕНБІ, СӘРСЕҢБИ, СӘРСЕНБІ, САРСЕҢБІ, СӘРСЕҢБІ}.

Potential replacement letters are determined by the dictionary (Fig. 3). A set of Kazakh words was obtained, formed by combinations based on the correspondence of letters in the dictionary. The set consists of options. Which one is the correct word is determined by comparing it to the database.

Replace number to letter – we replace incorrectly typed numbers and symbols with letters according to the algorithm. The solution to the error when numbers were printed instead of letters is implemented as a separate function (replace_number_to_letter(word)). The function algorithm is shown in Fig. 4. The error word is passed as a function parameter value. Letters of the Kazakh language {ә, і, қ, ғ, ү, ұ, ө, h} correspond to numbers and symbols {2, 3, 0, 5, 8, 9, –, =} on the keyboard in the dictionary as a key-value and are stored as dict_number_to_letter = {'2': 'ә', '3':'і', '4': 'ң', '5':'ғ', '8':'ү', '9': 'ұ', '0': 'қ', '–':'ө', '=':'h'}. Next, each character of the misspelled word is checked for a number, "–" or "=".

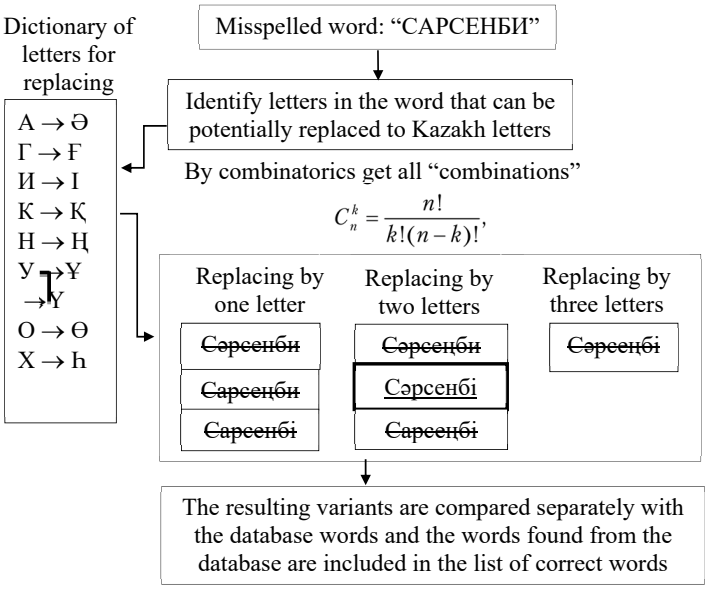

Fig. 3. Correction of words typed in Russian letters

If the symbol of the misspelled word is a number, that numeric value is replaced by the corresponding letter in the dictionary.

If the character of the misspelled word is equal to the sign "–", the first character "–" is replaced by the letter "ө" and its variant is included in the list of new words. The second character "ө" changes back to "–" and the cycle continues. The reason for getting one version with the replacement of the letter "ө" and the character "–" and the second version with the unchanged character is that the letter "ө" is entered incorrectly in a hyphenated word. Then, if the – character occurs in the word, this condition returns two words.

If there is a "=" sign in the misspelled word, it will be replaced by the letter corresponding to the key from the dictionary.

At the end of the algorithm, the modified word versions are returned as a list.

Words generated by the *replace to Kazakh letter* and *replace number to letter* algorithms are taken as candidate words and are recognized as correct when compared with the database (dictionary). The candidate word is the predicted correct version of the misspelled word. If there are several candidate words, similar words are usually taken. If their correction distance is different, you can immediately find the correct word. If the revision distance is the same, we get the word that is used more often in terms of probability.

The probability of a word is determined by the Bayes' rule [27]. According to the Bayes' rule, the frequency of each word is taken from the dictionary and the probability of its occurrence in the entire corpus is calculated. The probabilities are stored in a separate dictionary. Whenever a new word is added to the corpus, the probabilities are recalculated and the probability dictionary is updated.

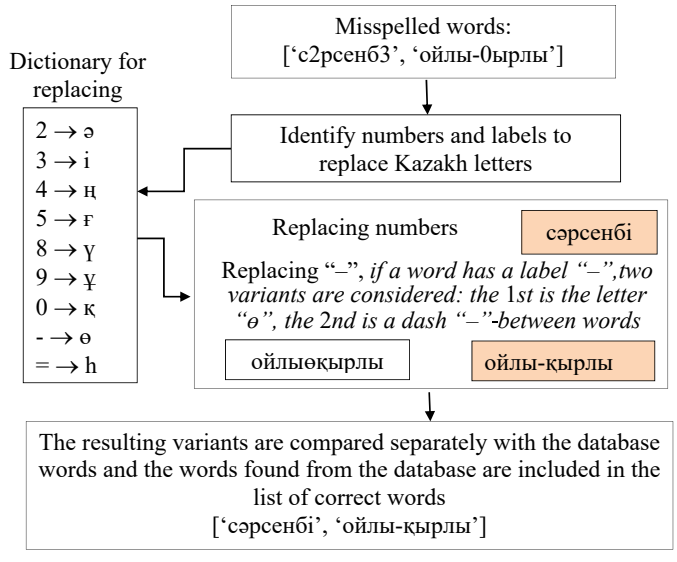

Fig. 4. Replacing numbers and signs – = in the text with Kazakh letters

5. 2. Algorithm of spelling error correction in Kazakh texts

The spelling error correction method for Kazakh texts is implemented by augmenting the Damerau-Levenshtein method including internal algorithms. The complete block diagram of the algorithms for spelling error correction in Kazakh texts is shown in Fig. 5. The work of the algorithm for correcting errors in the words of the Kazakh text begins with the source text. The source text is divided into words, and each word is given a separate loop. Next, the correctness of the word is checked in the database, if it is correct, it is added to the list of correct words, if it is misspelled, then it is corrected. The same sequence of steps and algorithms is used to correct all misspelled words. The proposed algorithm consists of several sub-algorithms, each algorithm is aimed at correcting various possible errors. Algorithms with a smaller number of corrective operations are started first, more complex algorithms are gradually used. Thus, you can save computing resources and avoid redundant operations.

Correcting a misspelled word starts with checking if the keyboard has changed from English to Kazakh. The misspelled word is replaced by the Kazakh keyboard and checked against the database. If the checked word is correct, it is added to the correct word set.

Otherwise, it goes to the block of distance correction algorithms. There, the correction is first performed using the *replace number to letter* and *replace to Kazakh letter* algorithms. After each algorithm, the results are checked against the database and, if they are correct, are added to the correct word set.

If the required word is not found in the database, it is checked whether the word is written in Latin letters. At the same time, the Kazakh word written in Latin letters is replaced by the corresponding Kazakh Cyrillic alphabet. The result is checked against the database, if the word is correct, it is added to the correct word set. If the word is written in non-Latin letters or the desired word is not found, it goes to the next algorithm.

If the correct version of the word is not found after the above algorithms, the correction range is further expanded, possible variants are generated by the insert, delete, switching adjacent letters and replace letters algorithms. But in this case, only substitutions of up to 2 letters are performed. This is because the complexity of the algorithm increases as the number of letters to be corrected increases. The resulting versions are checked against the database, and the word in the database remains as candidate words. If there are several candidate words, the correct word is obtained by statistical probability (the probability of the word appearing in the corpus). If no candidate words are found, then the original misspelled word is added to the words set and returned as is.

All corrected words are combined into one line and returned as text. Further, the correction algorithm is offered to the user in accordance with the type of program/application used.

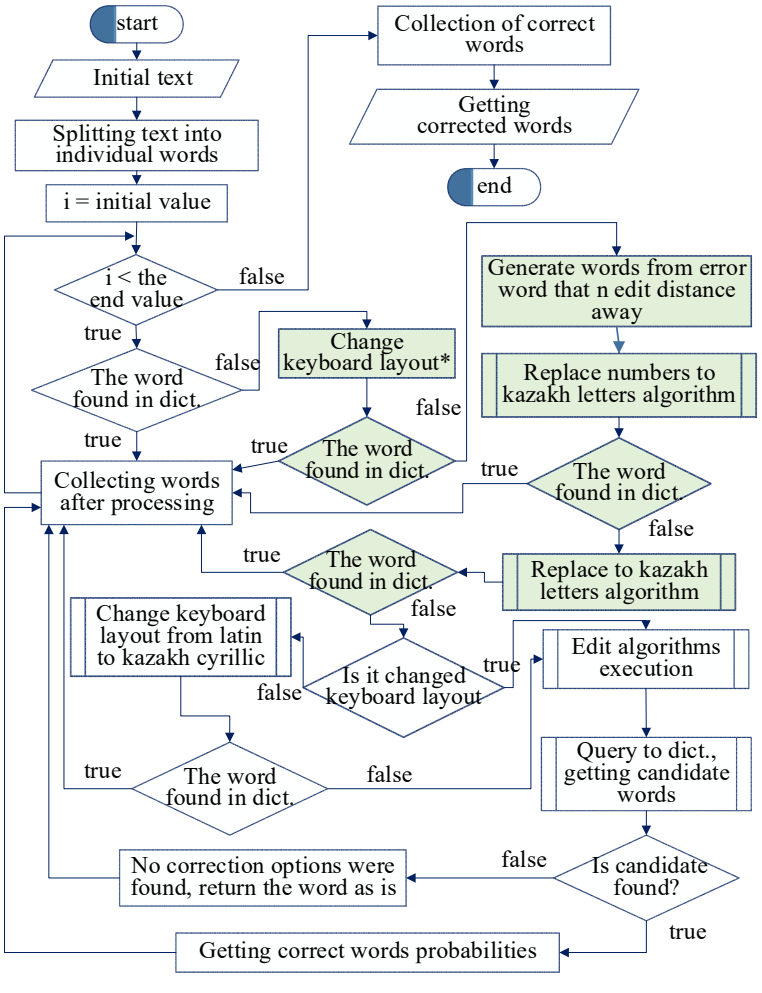

Fig. 5. Algorithm for correcting misspelled words in Kazakh texts

5. 3. Experimental testing of the developed algorithms

Collection of data for the dictionary. The dictionary is a key part of our proposed approach, as it is used to check spelling errors and predict the correctness of candidate words.

Therefore, we have collected a Kazakh language dictionary resource to correct spelling errors.

For compiling the dictionary, the frequency dictionary of the Kazakh language [28], 4 literary books and 5 technical textbooks were used. Among them, 30,515 unique words were taken from the frequency dictionary of the Kazakh language and supplemented with other textbooks. The vocabulary consists of a total number of 216,479 words, including 46338 unique words.

For the test, 500 words only with specific errors (Kazakh letters written with alternative russian letters, and numbers were typed instead of some Kazakh letters) were collected from the Telegram messenger. 500 words with typographical errors were prepared for correcting omission of letters, overwriting of letters and with incorrect writing by other letters.

Experimental testing. Two tests were carried out to check the developed algorithms: the first one was for correcting errors made by typing alternative Russian letters instead of Kazakh letters, the second was general typographical errors for correcting omission of letters, overwriting of letters and with incorrect writing by other letters. The tests were performed using the supplemented Damerau-Levenshtein method and the standard Damerau-Levenshtein method, and the results were compared.

Table 1 shows the results of tests for correcting specific spelling errors in Kazakh words. The total number of words with spelling errors is 500. The standard Damerau-Levenshtein method was able to correct 382 wrong words, and the augmented Damerau-Levenshtein method was able to correct 486 wrong words. The number of uncorrected wrong words is 118 in the standard Damerau-Levenshtein method, and 14 in the augmented Damerau-Levenshtein method. During the first test, wrong words were not recognized as correct words. As a result of the test, the Damerau-Levenshtein method showed 76.4 % accuracy, and the augmented Damerau-Levenshtein method showed 97.2 % accuracy. The results for F1-measure are 86.60 % and 98.5 %, respectively. It can be seen that the augmented Damerau-Levenshtein method showed better results, 1.5 % of wrong words were not corrected. In the standard method, it is 13.4 %.

Table1

Test for correcting words only with specific errors

Measurements	Damerau-Levenshtein method	Augmented Damerau-Levenshtein method
Total number of misspelled words	500	500
Correct word identified as a misspelled word (TN)	0	0
Number of corrected words (TP)	382	486
Number of uncorrected misspelled words / Candidate words not found or wrong candidate (FN)	118	14
Misspelled word identified as a correct word (FP)	0	0
Recall rate	76.4 %	97.2 %
Precision rate	100 %	100 %
Accuracy	76.4 %	97.2 %
F-measure	86.60 %	98.5 %

Table 2 shows the results of the spelling errors words omission of letters, overwriting of letters and with incorrect writing by other letters test. For this test, only words with typographical errors were taken and the number of wrong letters in the words is 1 or 2.

In the second test, the number of words with spelling errors is 500. The standard Damerau-Levenshtein method was able to correct 461 incorrect words, and the augmented Damerau-Levenshtein method was able to correct 464 incorrect words. The number of uncorrected misspelled words is 118 in the standard Damerau-Levenshtein method, and 14 in the augmented Damerau-Levenshtein method. During the test, 7 misspelled words were recognized as correct words. As a result of the test, the Damerau-Levenshtein method showed 92.2 % accuracy, and the augmented Damerau-Levenshtein method showed 92.8 % accuracy. F1-measure results are 95.9 % and 96.2 %, respectively. The results of the first type errors correction are similar in both methods, the difference between them is only 0.3 %.

Table 2

Words with typographical errors up to 2 letters

Measurements	Damerau-Levenshtein method	Augmented Damerau-Levenshtein method
Total number of misspelled words	500	500
Correct word identified as a misspelled word (TN)	0	0
Number of corrected words (TP)	461	464
Number of uncorrected misspelled words / Candidate words not found or wrong candidate (FN)	32	29
Misspelled word identified as a correct word (FP)	7	7
Recall rate	93.5 %	94.1 %
Precision rate	98.5 %	98.5 %
Accuracy	92.2 %	92.8 %
F1-measure	95.9 %	96.2 %

The subtask of error correction is to suggest correct variants from candidate words. In the best case, the exact correct word should be at the top of the suggestions. Table 3 shows the order of suggestion of correct words by the augmented Damerau-Levenshtein method and the standard Damerau-Levenshtein method. The correct candidate word suggestion version is at the top: 81.2 % in the augmented method and 60.8 % in the standard method. Occurrence among the first five candidate words is 90.2 % and 68.4 %, respectively. The presented experimental data are given according to the specific error correction test.

The developed algorithm and application detect a text error and suggest the correct option for misspelled words. If by processing the misspelled word only one correct version is identified, only that version is given. If there are more than one possible correct options, by reverse sorting from the most likely option downwards, the candidate words are suggested.

Table 3

Ordering the suggestion of correct words

The order of suggestions	Augmented Damerau-Levenshtein method	Standard Damerau-Levenshtein method
Correct suggestion in top one	81.2 % (406)	60.8 % (304)
Correct suggestion in top five	90.2 % (451)	68.4 % (342)
Correct suggestion in top ten	93.4 % (467)	71.6 % (358)
The presence of the correct version in candidate words	98.5 % (486)	76.4 % (382)

6. Discussion of the results of research on the development of a spelling error correction algorithm for Kazakh texts

It is known that creating an effective method of solving spelling errors is directly related to taking into account the features of natural language. As a result of the analysis, it was decided to divide the errors in Kazakh texts into two types: common spelling errors and specific spelling errors. Analyzing algorithms for correcting errors in different languages, it was noticed that they cannot correct specific spelling errors in the Kazakh text. In this regard, taking into account the features of the Kazakh language, it was proposed to develop algorithms for correcting spelling errors in the Kazakh language and create a test application to the performance of the algorithms. In addition, as a result of reviewing the conducted research work, it was decided to solve the problem of correcting errors in the text in two stages: identifying misspelled words and finding ways to correct them. In the reviewed papers, the Levenshtein [20], Damerau-Levenshtein [4], Jaro-Winkler [17], and probability methods [23] were often used to correct misspelled words. However, the best results are obtained by combining several methods [11, 29]. Therefore, when creating a text error correction model (Fig. 1), internal structures given in Fig. 3, 4 that correctly solve individual tasks were included in order to achieve a more accurate result. In the substructures, the combinatorics formula was used to generate all possible words from misspelled words. A list of possible correct words was compiled from the generated words, and probability theory was used to predict the most correct words. Based on this model, a complete algorithm for spelling error correction for Kazakh texts was developed (Fig. 5). For testing the performance of the algorithm, experimental tests were carried out. The results of experimental tests (Table 1) showed that the accuracy of specific error correction was 97.2 %, and the F1-measure was 98.5 %. According to the results of experimental tests, the augmented Damerau-Levenshtein method shows better results than the standard Damerau-Levenshtein method (Tables 1, 2) when correcting the second type of errors. The performance of both methods is similar in correcting common errors (Table 2), accordingly, the accuracy of the standard Damerau-Levenshtein method is equal to 92.2 % and the accuracy of the augmented Damerau-Levenshtein method is equal to 92.8 %.

The developed algorithm can be used to correct spelling errors in separate applications, text editors, managers, e-mails, etc. The result of the algorithm's work allows users to present a list of expected correct options when writing words with errors. Users can select a suitable variant from a list of suggested correct words. In addition to the aforementioned applications, text error correction is one of the primary processing stages of natural language processing tasks. For example, text processing tasks such as syntematic analysis of texts, clustering, predicting the next word, creating a question-and-answer system. Also, such developments can be used for handwriting recognition to correct words with incorrectly recognized letters [30]. If the text error is found accurately and correctly, this is a prerequisite for obtaining good results in the subsequent stages of processing.

A limitation of the proposed algorithm is that the complexity of the algorithm increases as the number of alternative letters or numbers and symbols to be replaced increases in the word identified as wrong. Also, a limitation is the computing power of the equipment on which the text processing application with the spelling error correction algorithm will run. The developed algorithm uses a dictionary of Kazakh words to identify misspelled words, so additional memory resources are required. This limitation is faced by all text error correction algorithms and applications that use a dictionary (or corpus).

The disadvantage of the algorithm is that when the statistical probability of the actual correct word among the proposed candidate words is lower than the statistical probability of another candidate word, the predicted correct version does not always match the actual correct word. If it is necessary to provide only one correct variant at once, the predicted correct variant with a high statistical probability is obtained. This situation may occur when the algorithm is used in automatic word prediction or text prediction systems, search query processing. These systems automatically select the first version of candidate words. This shortcoming can be corrected by applying machine learning where the algorithm suggests candidate words.

Further research should focus on improving the model, and an algorithm is needed to increase the probability of matching the actual correct word with the predicted correct word. For this, it is expected to use machine learning algorithms by taking into account neighboring words. In addition, the second future research will focus on investigating next word prediction by combining deep learning with current word spelling correction.

7. Conclusions

1. A model of correcting all types of spelling errors found in Kazakh texts was created. In order to have the possibility of correcting all types of errors found in the Kazakh text, the created model, internal structures and algorithms for correcting each type of error were developed and included in the common model. Also, when developing a model for correcting textual errors, combined methods were used to more accurately determine the correct version of the misspelled word.

2. Special algorithms have been developed to correct all specific spelling errors in Kazakh texts. The developed algorithms can be considered as a way to correct spelling errors in the Kazakh language. In order to take into account all types of spelling errors during text processing, the developed special algorithms were integrated with common Damerau-Levenshtein error correction algorithms. As a result, an extended version of the Damerau-Levenshtein method was proposed. As well as a complete block diagram of algorithms for correcting spelling errors in Kazakh texts is created.

3. According to the developed model and algorithm, a test application was created and tested for correcting misspelled words. The tests were conducted on the improved Damerau-Levenshtein method, developed by inserting algorithms for correcting specific errors of Kazakh words, and the usual Damerau-Levenshtein methods. The test results were compared and given in the form of tables (Tables 1–3).

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP09057872).

Data availability

Data will be made available on reasonable request.

References

1. Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. Communications of the ACM, 7 (3), 171–176. doi: https://doi.org/10.1145/363958.363994
2. Kwon, S., Lee, G. G. (2023). Self-feeding training method for semi-supervised grammatical error correction. Computer Speech & Language, 77, 101435. doi: https://doi.org/10.1016/j.csl.2022.101435
3. Sheng, L., Xu, Z., Li, X., Jiang, Z. (2023). EDMSpell: Incorporating the error discriminator mechanism into chinese spelling correction for the overcorrection problem. Journal of King Saud University - Computer and Information Sciences, 35 (6), 101573. doi: https://doi.org/10.1016/j.jksuci.2023.101573
4. Nagata, R., Takamura, H., Neubig, G. (2017). Adaptive Spelling Error Correction Models for Learner English. Procedia Computer Science, 112, 474–483. doi: https://doi.org/10.1016/j.procs.2017.08.065
5. Zukarnain, N., Abbas, B. S., Wayan, S., Trisetyarso, A., Kang, C. H. (2019). Spelling Checker Algorithm Methods for Many Languages. 2019 International Conference on Information Management and Technology (ICIMTech). doi: https://doi.org/10.1109/icimtech.2019.8843801
6. Kartbayev, A., Mamyrbayev, O., Khairova, N., Ybytayeva, G., Abilkaiyr, N., Mussayeva, D. (2021). Correction of Kazakh synthetic text using finite state automata. Journal of Theoretical and Applied Information Technology, 99 (22), 5559–5570. Available at: http://www.jatit.org/volumes/Vol99No22/29Vol99No22.pdf
7. Sorokin, A., Shavrina, T. (2016). Automatic spelling correction for Russian social media texts. Conference: Dialogue, International Conference on Computational Linguistics. Moscow, 688–701. Available at: https://www.researchgate.net/publication/303813582_Automatic_spelling_correction_for_Russian_social_media_texts
8. Song, X., Min, Y. J., Da-Xiong, L., Feng, W. Z., Shu, C. (2019). Research on Text Error Detection and Repair Method Based on Online Learning Community. Procedia Computer Science, 154, 13–19. doi: https://doi.org/10.1016/j.procs.2019.06.004
9. Abdellah, Y., Lhoussain, A. S., Hicham, G., Mohamed, N. (2020). Spelling correction for the Arabic language space deletion errors-. Procedia Computer Science, 177, 568–574. doi: https://doi.org/10.1016/j.procs.2020.10.080
10. Kumar, R., Bala, M., Sourabh, K. (2018). A study of spell checking techniques for Indian Languages. JK Research Journal in Mathematics and Computer Sciences, 1 (1), 105–113. Available at: http://jkhighereducation.nic.in/jkrjmcs/issue1/15.pdf
11. Chaabi, Y., Ataa Allah, F. (2022). Amazigh spell checker using Damerau-Levenshtein algorithm and N-gram. Journal of King Saud University - Computer and Information Sciences, 34 (8), 6116–6124. doi: https://doi.org/10.1016/j.jksuci.2021.07.015
12. Goslin, K., Hofmann, M. (2022). English Language Spelling Correction as an Information Retrieval Task Using Wikipedia Search Statistics. Proceedings of the Thirteenth Language Resources and Evaluation Conference. Marseille, 458–464. Available at: https://aclanthology.org/2022.lrec-1.48/
13. Gowri, S., Sathish Kumar, P. J., Geetha Rani, K., Surendran, R., Jabez, J. (2022). Usage of a binary integrated spell check algorithm for an upgraded search engine optimization. Measurement: Sensors, 24, 100451. doi: https://doi.org/10.1016/j.measen.2022.100451
14. Gupta, P. (2020). A Context-Sensitive Real-Time Spell Checker with Language Adaptability. 2020 IEEE 14th International Conference on Semantic Computing (ICSC). doi: https://doi.org/10.1109/icsc.2020.00023
15. Makazhanov, A., Makhambetov, O., Sabyrgaliyev, I., Yessenbayev, Z. (2014). Spelling Correction for Kazakh. Lecture Notes in Computer Science, 533–541. doi: https://doi.org/10.1007/978-3-642-54903-8_44
16. Yanfi, Y., Gaol, F. L., Soewito, B., Warnars, H. L. H. S. (2022). Spell Checker for the Indonesian Language: Extensive Review. International Journal of Emerging Technology and Advanced Engineering, 12 (5), 1–7. doi: https://doi.org/10.46338/ijetae0522_01
17. Friendly, F. (2019). Jaro–Winkler Distance Improvement For Approximate String Search Using Indexing Data For Multiuser Application. Journal of Physics: Conference Series, 1361 (1), 012080. doi: https://doi.org/10.1088/1742-6596/1361/1/012080
18. Kantrowitz, M., Baluja, S. (2003). Pat. No. US6618697B1. Method for Rule-Based Correction of Spelling and Grammar Errors. Available at: https://patents.google.com/patent/US6618697B1/en
19. Sarkar, D. (2016). Text Analytics with Python. Apress Berkeley, 385. https://doi.org/10.1007/978-1-4842-2388-8

20. Ceska, Z., Hanak, I., Tesar, R. (2007). Teraman: A Tool for N-gram Extraction from Large Datasets. 2007 IEEE International Conference on Intelligent Computer Communication and Processing. doi: https://doi.org/10.1109/iccp.2007.4352162
21. Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics-Doklady, 10 (8), 707–710. Available at: https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf
22. Rauf, S. A., Saeed, R., Khan, N. S., Habib, K., Gabrail, P., Aftab, F. (2017). Automated Grammatical Error Correction: a Comprehensive Review. NUST Journal of Engineering Sciences, 10 (2), 72–85. Available at: https://journals.nust.edu.pk/index.php/njes/article/view/219
23. Jurafsky, D., Martin, J. H. (2023). Spelling Correction and the Noisy Channel. Speech and Language Processing. Available at: https://web.stanford.edu/~jurafsky/slp3/B.pdf
24. On the transfer of Kazakh writing from Latinized to a new alphabet based on Russian (1981). Collection of laws of the Kazakh USR and decrees of the Presidium of the Supreme Soviet of the Kazakh USR, 1, 1938–1981.
25. The development of Kazakh Soviet linguistics (1980). Publishing house "Science" of the Kazakh USR, 128–242.
26. Kazsur 903-90. Computer facilities. Keyboards. The location of the letters of the Kazakh alphabet (2023). Available at: https://online.zakon.kz/Document/?Doc_id=1045019&pos=1;-16#pos=1;-16
27. Norvig, P. (2016). How to Write a Spelling Corrector. Available at: https://norvig.com/spell-correct.html
28. Zhubanov, A. K., Zhanabekova, A. A., Karbozova, B. D., Kozhakhmetov, A. K. (2016). Frequency dictionary of the Kazakh language. Almaty, 792.
29. Desta, S. G., Lehal, G. S. (2023). Automatic spelling error detection and correction for Tigrigna information retrieval: a hybrid approach. Bulletin of Electrical Engineering and Informatics, 12 (1), 387–394. doi: https://doi.org/10.11591/eei.v12i1.4209
30. Yeleussinov, A., Amirgaliyev, Y., Cherikbayeva, L. (2023). Improving OCR Accuracy for Kazakh Handwriting Recognition Using GAN Models. Applied Sciences, 13 (9), 5677. doi: https://doi.org/10.3390/app13095677