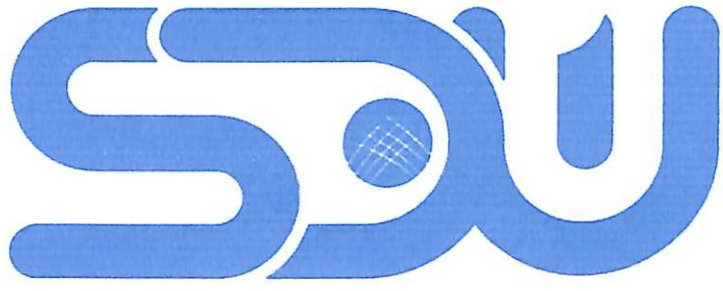
Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University



Zhazira Koppagambetova

Correcting text in morphologically rich languages

A thesis submitted for the degree of
Master in Computer Science
(degree code: 7M061002)

Kaskelen, 2022

Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University
Faculty of Engineering and Natural Sciences

Correcting text in morphologically rich languages

A thesis submitted for the degree of
Master in Computer Science
(degree code: 7M061002)

Author: **Zhazira Koppagambetova**

Supervisor: **MSc. Ardak Shalkarbay-uly**

Dean of the faculty:
Associate Professor, PhD Azamat Zhamanov

Kaskelen, 2022

Suleyman Demirel University

Faculty of Engineering and Natural Sciences

Department of Computer Science

Dean of Faculty

Associate Professor, PhD Zhamanov A.

« 8 » June 2022

Topic of the thesis:

Correcting text in morphologically rich languages

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science", SDU, 2020-2022

Head of Department: Associate Professor, PhD Cemil Turan

Academic Supervisor: MSc. Ardak Shalkarbay-uly

Master's student: Zhazira Koppagambetova

Kaskelen, 2022

Abstract

Currently, there is a tendency to increase the volume of documents containing complex text structures. Professional text proofreaders provide their services to correct errors in the text for not a small amount of money. A very large number of people every day around the world, who are closely connected with science or education, write a large volume of articles and, accordingly, each and all of them should be written without errors and in the correct version. In order to automate this process, it was necessary to develop an algorithm based on the methods of analysis and correction of the input text. For high-quality text synthesis, it is necessary to use machine learning technologies that require deep knowledge and understanding in this area. Among the many machine learning algorithms Hunspell algorithm seems to be one of the best to solve this issue. The essence of this algorithm is to bring all the words contained in the text to the original format. Thus, this work is based on multilevel segmentation of errors of Kazakh-language text from the Internet or by manual user input. It is worth noting that due to technological progress, the main source of linguistic research is social networks, which is a critical problem due to the dubious fidelity of texts. The main purpose of this work was the formation and development of a spell-checking algorithm for the Kazakh language based on the existing Hunspell algorithm for English. As a result, the Hunspell algorithm was studied, where the methods of extracting the base of the word, as well as the ways of aggregation of the normal form were analyzed. The essence of this algorithm lies in the inherent rule for detecting errors and typos in a word. These rules are contained in the dictionary and its dependence on the algorithm is directly proportional. Thus, the more rules there are, the more errors the algorithm detects and the more accurate it is. In addition, the forms and categories of Kazakh words and their differences from English words were also considered. As a result, an algorithm was created to check the spelling of Kazakh words based on the existing Hunspell algorithm. The significance of this work seems to be that the correct spelling of words, sentences forms a literate society, which is key in the development of modern civilization.

Keywords: Kazakh language, natural languages, processing, Hunspell algorithm, text processing, text correction

Аңдатпа

Қазіргі уақытта мәтіндердің күрделі құрылымдары бар құжаттар көлемінің ұлғаю үрдісі байқалады. Кәсіби мәтіндік редакторлар мәтіндегі қателерді түзету үшін өз қызметтерін аз мөлшерде ұсынады. Ғылыммен немесе біліммен тығыз байланысты бүкіл әлем бойынша күн сайын өте көп адамдар көптеген мақалалар жазады және сәйкесінше олардың әрқайсысы қатесіз және дұрыс нұсқада жазылуы керек. Бұл процесті автоматтандыру үшін енгізілген мәтінді талдау және түзету әдістеріне негізделген алгоритм жасау қажет болды. Мәтіндерді сапалы синтездеу үшін осы салада терең білім мен түсінуді қажет ететін Машиналық оқыту технологияларын қолдану қажет. Hunspell машиналарын оқытудың көптеген алгоритмдерінің ішінде алгоритм осы мәселені шешудің ең жақсы әдістерінің бірі болып көрінеді. Бұл алгоритмнің мәні мәтіндегі барлық сөздерді бастапқы форматқа келтіру болып табылады. Осылайша, бұл жұмыс интернеттен қазақ тіліндегі мәтіннің қателерін көп деңгейлі сегментациялауға немесе қолданушыны қолмен енгізуге негізделген. Технологиялық прогреске байланысты лингвистикалық зерттеулердің негізгі көзі әлеуметтік желілер болып табылады, бұл мәтіндердің күмәнді сенімділігіне байланысты маңызды мәселе болып табылады. Бұл жұмыстың басты мақсаты Hunspell ағылшын тіліне арналған қолданыстағы алгоритм негізінде қазақ тіліне арналған орфографияны тексеру алгоритмін қалыптастыру және әзірлеу болды. Нәтижесінде Hunspell алгоритмі зерттелді, онда сөздің негізін алу әдістері, сондай-ақ қалыпты форманың агрегация жолдары талданды. Бұл алгоритмнің мәні сөздегі қателер мен терулерді анықтаудың негізгі ережесінде жатыр. Бұл ережелер сөздікте бар және оның алгоритмге тәуелділігі тікелей пропорционалды. Осылайша, ережелер неғұрлым көп болса, алгоритм соғұрлым көп қателерді анықтайды және дәлірек болады. Сонымен қатар, Қазақ сөздерінің формалары мен категориялары және олардың ағылшын сөздерінен айырмашылықтары қарастырылды. Нәтижесінде, қолданыстағы Hunspell алгоритмі негізінде Қазақ сөздерінің емлесін тексеру алгоритмі құрылды. Бұл жұмыстың маңыздылығы сөздердің, сөйлемдердің дұрыс жазылуы қазіргі өркениеттің дамуында шешуші болып табылатын сауатты қоғамды қалыптастырады.

Түйін сөздер: қазақ тілі, табиғи тілдер, өңдеу, Hunspell алгоритм, мәтінді өңдеу, мәтінді түзету

Аннотация

В настоящее время прослеживается тенденция увеличения объема документов, содержащих сложные структуры текстов. Профессиональные текстовые корректоры предоставляют свои услуги по исправлению ошибок в тексте за не малую сумму денег. Очень большое количество людей с каждый днем по всему миру, которые тесно связанны с наукой илиобразованием, пишут большой объем статей и соответственно каждая и все из них должны быть написаны без ошибок и в корректномварианте. Для того чтобы сделать автоматизацию данного процесса необходимо было разработать алгоритм, опирающееся на методыанализа и корректировки вводимого текста. Для качественного синтеза текстов необходимо применение технологий машинного обучения, требующих глубоких знаний и понимания в данной области. Среди множества алгоритмов машинного обучения Hunspell алгоритм представляется одним из лучших для решения данного вопроса. Суть данного алгоритма заключается в приведении всех слов, содержащихся в тексте, к изначальному формату. Таким образом, данная работа основана на многоуровневой сегментации ошибок казахско-язычной литературы из Интернета. Стоит отметить, что в связи с технологическим прогрессом, основным источником лингвистический исследований являются социальные сети, что является критичной проблемой из-за сомнительной верности текстов. Главной целью данной работы было формирование и разработка алгоритма проверки орфографии для казахского языка на базе существующего алгоритма для английского языка Hunspell. Как следствие, был изучен алгоритм Hunspell, где были проанализированы методы извлечения основы слова, а также пути агрегации нормальной формы. Суть данного алгоритма заключается в заложенном правиле обнаружения ошибок и опечаток в слове. Данные правила содержатся в словаре и зависимость у него с алгоритмом прямо пропорциональна. Таким образом, чем больше правил, тем больше ошибок обнаруживает алгоритм и тем точнее он. Помимо этого, были также рассмотрены формы и категории казахских слов и их отличия от английских слов. По итогу, был создан алгоритм для проверки правописания казахских слов на базе существующего алгоритма Hunspell. Значимость данной работы представляется в том, что правильность написания слов, предложений формирует грамотное общество, которая является ключевой в развитии современной цивилизации.

Ключевые слова: казахский язык, естественные языки, обработка, Hunspe
алгоритм, обработка текста, коррекция текста

Acknowledgements

I express my deep gratitude for the help in the process of preparing the thesis to my supervisor – Ardak Shalkarbay-uly. His pedagogical and scientific approach, excellent knowledge of the subject inspired me to do this work. I was strongly impressed by his ability to guide the student, carefully check the results of the work, give detailed and fair comments, and at the same time his desire to give the undergraduate freedom in the implementation of ideas. All the time from choosing a topic to completing the work, he helped with recommendations and provided moral support. He was a great help to me in my work, not only helped me to correct mistakes in my work, but also gave me valuable advice. Let me express my sincere gratitude to my reviewers for the work done, recommendations and comments, as well as to the members of the attestation commission, the leadership of the Suleyman Demirel University and to all my teachers. Heartfelt thanks to my family and friends for their support and help.

Contents

Chapter 1

Introduction

1.1 Motivation

The current study focuses on correcting text in morphologically rich languages. The greatest invention of the people is writing. There are several complex issues involved in the study of the origins of writing in modern linguistics. Among these is the idea that ancient writing predates the development of modern writing. Since the middle ages, Central Asia and Kazakhstan have been speaking written languages. A significant part of Eurasia's history has been played by Kazakhs and other peoples descended from ancient Turkic peoples.

Spelling is one of the most important components of national culture, and the presence of a generally binding set of spelling rules is one of the signs of the cultural health of society. Kazakh orthography was formed in the process of a long historical development, so there are quite a lot of spellings in it that no longer correspond to either its basic principle or the current state of affairs. The spelling system of the Kazakh language is determined by a set of principles, the main of which is morphological.

Spell checking systems detect a significant number of errors and typos. The larger the dictionary of the system, the more rules and verification algorithms it contains, the greater the percentage of errors detected by it. But no spell-checking system can guarantee the complete absence of errors and typos in the document. The spell-checking systems used in most modern text editors make it possible to identify a significant part of the typos and spelling mistakes made by the user. The principle of operation of a typical spell-checking system is as follows: the built-in dictionary of the system contains a large set of words of the

analyzed language in various grammatical forms (time, number, etc.), the system tries to find the word being checked in this dictionary. If a word is found, it is considered to be spelled correctly. If a word is not found in the dictionary, but there are similar words, an error message is displayed and possible replacement options are offered. If nothing similar is found, the system suggests correcting the word or entering it into the dictionary. Of course, the principle of automated spell checking is set out here in a very simplified form, but its essence is exactly that.

The modern spelling norm requires knowledge of, firstly, more than a hundred spelling rules, secondly, a large number of exceptions to the rules and, thirdly, the spelling of so-called dictionary words, i.e. words whose spelling is not regulated by the rules. It is obvious that the Kazakh spelling system is objectively complex. Active processes in the field of vocabulary, replenishment of the literary language with new words, mainly borrowed ones, introduce additional difficulties. It is impossible to know how to spell all the words that exist in the language. It is very important to know the regulatory sources to which you need to turn when a particular problem arises. Since there is currently a decline in literacy, the culture of writing, the topic of this course work is relevant. The object of research in the course work is the correction of the text in the Kazakh language. The purpose of this course work is to refine the existing methods and algorithms for correcting the text in the Kazakh language.

A general direction of artificial intelligence and mathematics is Natural Language Processing (NLP). A number of applied tasks are examined involving information technologies for the analysis and synthesis of natural languages. The term "analysis" refers to understanding the language, while the term "synthesis" refers to generating a literate text. It will require a more convenient form of computer interaction and human interaction to solve these problems. A variety of applications use NLP today, including voice assistants, automated text translations, and text filtering. Research and development of various tasks are carried out in the field of NLP. Among these tasks, the following can be distinguished:

- Text recognition, speech, speech synthesis;
- Morphological analysis (words);
- Collection and storage of linguistic resources (corpus, dictionaries, the-

saurus, etc.);

- Syntactic analysis, tokenization of sentences;
- Extraction of relationships, definition of language, analysis of emotional coloring;
- Annotation of the document, translation, analysis of the subject of the text;
- Information search, machine translation, etc.

The field of natural language processing in world science is quite mature, many formal models, methods and algorithms have been developed. There are various high-level applied software systems for the analysis and processing of natural languages, collection and storage of linguistic data. The active integration of Kazakhstan into the world community and the increasing volume of information flows between our country and its foreign partners, the real need for various segments of the population in information technologies in everyday life is increasing.

Researchers and developers in the field of computational linguistics are gaining a lot of importance because the volume and type of natural language information on the Internet and social media is increasing rapidly. Processing of information resources in natural language requires the presence of text corpora and thesauri. To create and process them, markup languages and ontological models of subject areas are required. The existing markup languages mainly contain concepts of Romano-Germanic and Slavic language groups. The metalanguage is intended for marking up texts of natural languages.

A metalanguage has the following properties: with the help of its linguistic means, it is possible to express everything that is expressible by means of an object language, and to designate all signs, expressions of an object language for which there are names; in a metalanguage, it is possible to talk about the properties of an expression of an object language and the relations between them; it is possible to formulate definitions, designations, rules of formation and transformation for object language expressions. These metalanguages are not adapted to describe the Turkic languages, which have many concepts different from the concepts of the above language groups. Therefore, the creation of a single metalanguage for marking up texts of Turkic languages (UniTurk) is an urgent task for processing Turkic languages.

Such a language will allow to unify the markup, facilitate their understanding and use common software, as well as allow for a comparative analysis of linguistic concepts of the Turkic languages. And also, in addition to the markup language, for computer processing of any natural languages, the formalization of their grammatical (morphological and syntactic) rules is required, the development of algorithms for the analysis and synthesis of words and sentences according to these rules, the software implementation of all these algorithms, the creation of thesauri by subject areas, the construction of text corpora (a database of marked texts) and other programs for text analysis and processing. Ontology is used to formalize grammatical rules. Ontology is a conceptual scheme consisting of many concepts and many statements about these concepts, on the basis of which classes, relationships, properties, functions and individuals can be described. We can also say that ontology is a knowledge base, because if you add interpretive functions to the structural–semantic model, it will become a knowledge base. All ontological models are built in the Protégé environment, which makes it possible to simplify the process of creating, downloading, changing and transforming the knowledge base, as well as to provide it for general use in the form of joint viewing and editing.

The reasons why people are now using more programs that correct the text in the Kazakh language:

1. Remove grammatical, morphological and word-formation errors.
2. Check syntax, spelling and punctuation.
3. Make sure that the text is designed in accordance with the uniform spelling requirements.
4. Correct layout errors and evaluate the artwork of the future book.

Reading any document, attention is involuntarily drawn to the style of presentation, ease of perception, content and brevity of the narrative. However, it is not uncommon to encounter typos and errors in documents. Errors in the text can blur all the positive impression about the author, and sometimes cause serious damage to business. Communicating in their native language, it is almost always noticed that the author made a mistake in this place. In addition, it is usually

possible to guess what the author really meant. The situation is much more complicated in cases where there is communication with foreigners. A mistake in writing just one word can significantly distort the meaning of the entire message, and even intuition may not help the recipient of the text, since the language of communication for the recipient is not native. To correct the typed text, programs were created to check spelling, syntax, grammatical rules for constructing sentences, hyphenation, etc. The first and most active users of such programs were those who create and edit texts.

Information about subject areas is presented in the knowledge base in the form of semantic structures of facts, and these structures explicitly define standard relationships between objects. A fact is a localized minimal set of natural language concepts that reflect the relationship between classes of objects, classes and their elements, as well as between elements of different classes. Facts are also called judgments or statements. Classes of objects are listed in the database. The semantic structure of a fact is represented by a tree, in the nodes of which there are concepts, and the arcs reflect the semantic relations between them. At the same time, such a graphical structure is translated into a predicate form. A query that is formulated in a natural language with minor restrictions on syntax is transformed through a predicate form into a semantic structure similar to the structures of a knowledge base. After that, the structure of the query in the knowledge base is recognized. If there is a match, the structure of the knowledge base is transformed into a predicate, and the latter is translated into a response in natural language. The search engine supports composite and complex questions. It can not only duplicate the issue with the issuance of the required data, but also provide forecasts regarding resources, the implementation of the plan and other valuable information that is not explicitly presented

NLP is closely related to linguistics, and when teaching models and solving certain NLP tasks, it is certainly necessary to take into account the peculiarities of a particular language. For example, in Kazakh, unlike English, there is a free word order and a large number of word forms. This, of course, affects the quality of the resulting models and, perhaps, even the choice of algorithms. For example, for the Kazakh language, lemmatization is usually used to normalize words, that is, they bring all words from the text to their normal vocabulary form. So, for nouns it will be a word (кітап → кітабі, кітап → кітаптар, and so on). But for

English, such a method as stemming works well, in which the original word form is cut to its base (features → feature, accessible → access, and the like). In addition, the quality of models is influenced by the amount of data for training (as a rule, the more data, the better). The number of texts in the public domain varies for different languages. Fortunately, there are quite a lot of Kazakh-language texts at the moment, but still less than for English. Most text editing systems have a tool for automatic spelling error checking (when one or more letters are spelled incorrectly in a word). Their principle of operation: the program analyzes every word in the text and searches for the same in the Database of all words and their various forms.

Such a text check ensures that the words in the text will be spelled correctly (as in a dictionary), but does not protect against matching errors and syntactic errors in the sentence. For example, the sentence "Мен Отанымды өте сүйемін" is wrong, but the text editing system will not show the correct option: "Мен Отанымды оте сүйесін" The grammar checker helps to avoid such errors.

Many programs have been created that check spelling errors. But so far (as far as I know) there are no programs checking grammatical errors in a sentence (for inflectional/inflectional languages — in which words have many forms); that is, checking not only spelling errors, but also coordination and syntax errors.

1.2 Aims and Objectives

It has been reported that most grammar error correction research these days focuses on English, and that little is known about grammar error correction in other languages. This research project, is dedicated that Kazakh, as an example of a morphologically rich language, will be used to identify and correct common writing errors in Kazakh. In the absence of publicly available data, research on spell checking in the Kazakh language is difficult. Natural language processing is difficult in Kazakh due to its developed nominal and verb morphology and free word order. This makes it a suitable medium for testing a wide range of spelling correction models, even though future results can be applied to any other language with similar properties.

A Grammar Error Correction (GEC) program is designed to correct spelling and punctuation errors, as well as grammatical mistakes and the choice of words.

Among the many applications of a computerized spelling correction system is the correction of search queries, spellchecking within browsers, and text editors. Natural language processing began around the same time as the development of computerized spelling correction systems. As a result of the rapid growth of social networks, in contrast, has made spelling mistake correction a priority again. Over more than a decade, corpora of Web texts, including blogs, microblogs, forums, have become the primary sources of large-scale research. With only limited manual intervention, the system collects and processes very large corpora without human intervention.

The main objective is to develop the existing algorithms on Kazakh language in such collections, textual materials contain a variety of forms of varying spellings, including everything from mere typos and orthographic errors to dialectal and linguistic differences. Furthermore, grammatical errors are inevitable, since the more social media texts there are, the more likely they are to be written by writers with little formal education and, as a result, make errors. By increasing the proportion of out-of-vocabulary words in text, we can reduce the success of further NLP tasks from lemmatization to parsing and extraction of information. To conclude, it is important to identify and eliminate at least significant spelling errors in texts with a high degree of precision.

1.3 Thesis Outline

The study focuses on the development of existing algorithms for correcting texts. The typo generator accepts a word as input and gives a word with a certain number of errors as output. Errors can be of the following types: replacing one letter with another, inserting a new letter, deleting an existing one, as well as rearranging two letters in places. The probabilities of each type of error are configured separately, the total probability of making a typo (depending on the length of the word) and the probability of making a second typo are also configured. The refinement of spell-checking systems that show optimal results is one of the key problems in the direction of natural language processing (NLP). The relevance of this problem is confirmed by the widespread introduction of spell-checking systems into most types of software products (text editors, automatic translation systems, etc.), as well as the use of NLP and artificial intelligence by researchers

and developers for other tasks. A few years ago, there were no articles devoted to the task of reviewing spelling checking systems for the Russian language. In this regard, an urgent task is to finalize spelling verification systems for the Kazakh language among the existing solutions. The next chapter The object of this study is the spelling checking systems for the Kazakh language. The subject of the study is the refinement of the existing algorithms of the spelling check system for the Kazakh language. The purpose of the work is the selection and analysis of the spelling check system for the Kazakh language.

Chapter 2

Literature Review

2.1 Definition

Spell checking application determines whether a document contains mistakes in spelling, grammar, and punctuation. A typo or an inaccuracy is indicated by highlighting - usually it is done by using underlining. Occasionally, a user is offered not just the option of selecting one of the correct spellings, but also a comment describing the correct way to spell the text. Some software programs, including text editors, messaging clients, dictionary programs, or web browser extensions, provide spell-checking functionality as a stand-alone feature. This function can also be incorporated into standalone programs [15].

2.2 History

Modern automated proofreaders of texts using methods of morphological and syntactic analysis of sentences and allowing them to detect a wide range of spelling and a number of syntactic errors in texts in natural languages, nevertheless, they are not yet able to identify, much less correct, word compatibility errors [25]. However, in modern computational linguistics, ways to solve this problem are outlined, in particular, a method for correcting special lexical and semantic errors based on word usage statistics is proposed [30].

2.2.1 Introduction to the first spell-checking modern software

In the 1970s, the first spell-checking modern software was developed for mainframe computers [8]. Language experts at Georgetown University designed the original system of this type for IBM. In 1980, the first IBM PC packages appeared for the TRS-80 and CP/M. In 1981, these appeared for the CP/M [13]. In addition to offering solutions for the PC, there were also offerings for Unix, Apple Macintosh and VAX. Several OEM software companies flooded the market with OEM software products, such as Maria Mariani, Circle Noetics, Microlytics, Proximity, Soft-Art and Reference Software. In addition to the stand-alone programs for the PC, many of these verification systems were possible to run within word processing programs (using TSR mode if there was enough memory). It is unfortunate to say that these separate packages didn't last long, because the creators of the most popular modern word handling software (such as WordPerfect and WordStar) in the mid-1980s incorporated spelling checking systems into their applications, buying the tools mostly from these companies, which promptly added support for European languages and later Asian languages as well. However, this made the development of spell-checking more challenging, especially with respect to Finnish and Hungarian [39]. WordPerfect was eager to expand into new markets, despite the fact that the market for word processing software in Iceland was not paying off. Several web browsers have added spell-checking capabilities recently, including Firefox 2, Google Chrome, Opera and Konqueror. Even the e-mail instant messenger client Pidgin and client Kmail have implemented GNU Hunspell spell-checking mechanisms. In addition to character encoding, Hunter acts as both a spelling and morphological analyzer. As a point of interest, Fred J. Damerau [9], a Hungarian biologist and developer, developed Hunspell originally for Hungarian. Fred J. Damerau has been working on OpenOffice.org since 2002, where he is directly involved in improving the spelling and grammar checking functionality of Hunspell, non-standard words, and agglutinative language inputs. Furthermore, Hunspell is built on top of MySpell, which enables compatibility between the two dictionaries mentioned above. One notable difference between MySpell and Hunspell is that MySpell uses a single-byte character encoding, whereas Hunspell uses Unicode UTF-8.

These are the main characteristics of Hunspell dictionary

- Provides extensive support for languages with unique features
- Enhancement of suggestion using similarity in n-grams, rules, and dictionaries.
- Morphological analysis, stemming and generation. MySpell is the foundation of Huntspell, which makes them compatible with each other.
- GPL/LGPL/MPL licensed C++ library.

2.2.2 Types of spelling checkers

Examples of types of spelling checkers found in the literature. The error can be either orthographic or typographic. The first category of error, also called cognitive error, includes real-word error, phonetic error, grammar error, and author misconceived error [5]. The second type of error, also known as physical error, includes keyboard layout errors, typing errors, split errors, run-on errors, transmission errors, storage errors, and residual errors. Detailed explanations of each type of error follow [34]:

1. An orthographic (Cognitive) error
   (a) A real-word error is a mistake that results in a correct word, for example, writing picturesque instead of picaresque.
   (b) A phonetic error entails a misspelled word. Examples include fonetik, which means "phonetic."
   (c) Grammar rules are broken due to spelling errors, such as using RSX instead of RSX and capitalization mistakes.
   (c) Misconcei Miswritten error: Misspelling that appears and sounds the same as the proper term, for example, writing anonomy
2. Typographical (Physical) Error
   (a) Typing error: If you are typing too quickly or sluggishly there is a chance that you will misspell the words. In situations where the user

is typing quickly, letters can be dropped due to the speed, such as when typing amp for amplifying. The user often repeats letters due to sluggish typing speed, such as typing independent for independent.

(b) Typing abouy for about is an example of a keyboard layout error, where an adjacent key is entered instead of the correct actual key. Physical errors are caused by the following devices:

(c) A split error occurs when a single word is separated from another. For example, forgetting to forgot.

(d) An example of a run-on error is when two words are combined to make a single word, such as "of the to of the".

(e) Transmission error: For example, sending the number 0 as 1 instead of the number 1. Also, a 0 or a 1 can sometimes be returned when nothing was sent. This can happen when cat is sent, but the recipient receives at.

(f) A storage error occurs when a data store contains missing characters in a word or the word has been converted to another. Take cat for instance.

(g) The presence of a residue in a word can be caused by a residue error. An example would be cat catqA. Some residues may contain alphabetic characters or other unusual characters. In this article, misspelling errors are examined in the context of phonics, author misconceptions, typing, and keyboard layout.

2.3 Word-formation science

The grammatical approach to word formation refers to the common traditions of Russian and Kazakh linguistics. It dissolves word formation in morphology, reducing it mainly to the word formation of parts of speech. Only by the fifties of the XX century, Russian word formation receives the status of an independent science, having its own object in the form of special word-forming units and in the form of an autonomous system endowed with specific features. This point of view has become widespread, but has not removed discussions on the problem of level stratification of language. Word formation as a separate tier of language is not

even recognized by all derivatologists. More often it is considered as a sublevel [40]. This interpretation of word formation in modern linguistics is supported by the assessment of language levels from the standpoint of their functional significance, i.e. from the standpoint of their participation in speech activity, in which vocabulary and syntax are brought to the fore, ensuring the transmission of the content side of speech, phonetics- phonology and morphology, which formalize materializing speech, are in the second place.

2.3.1 Kazakh Linguistics

In Kazakh linguistics, word formation remains the least developed part of it, since its basic concepts, including the units of word formation, its means, models, and system are at the very initial stage of development. Therefore, the Kazakh word formation has not yet accumulated sufficient data to obtain "autonomy". Nevertheless, the presence of not only Russian, but also Kazakh word formation as a language area of its specific features, its system is demonstrated by the linguistic reality itself and confirms its comparative analysis. Thus, the comparison reveals that complexes of derived words representing the proper face of Russian word formation are inherent in the Kazakh language. The separation of word formation as an independent science has significant prospects in terms of deepening the understanding of the systemic organization of language and language (speech) processes, which include not only the construction of sentences, texts, but also the production of new words [17]. The current state of word-formation theory reflects the course of development of linguistics of the XX century and, first of all, indicates the spread of ideas of structural linguistics. It was their influence that determined the emergence of a systemic-structural, on the one hand, and a functional direction in Russian word formation, on the other hand. Within the framework of the first direction, the word-formation level is represented as a system of complex units organized by different word-formation relations. Within the framework of the second direction, word formation is considered as the process of the birth of derived words, motivated by the need to enclose new information in the language shell, which previously had no name in the language [18].

2.3.2 Diachrony of Kazakh Language

"Derived" words, which are the object of word formation, are mostly not produced, but reproduced. Their production, as a rule, refers to the diachrony of language. Word formation, on the other hand, adopted a synchronous approach, which was established in linguistics even before its isolation. The limited scope of synchrony imposed a ban on referring to the dynamics of the creation of derived words, which was assumed by the term "word formation" itself. A small number of neoplasms, contemporaries of native speakers, could not displace? the former object of science. Synchronous word formation received ready-made reproducible words related to statics, but at the same time preserving the dynamic side reflecting the process of their production. The possibility of distinguishing these two sides of derived words and getting out of the contradictory situation of word formation was obtained through the development of appropriate scientific directions in it [35]. Modern word formation considers ready-made words, firstly, to establish their connections with other derived and non-derived words in the system (system-structural approach), secondly, to restore the process of their formation (functional approach), thirdly, to determine the composition of morphemes in the word and their system in the language. Consequently, words formed with the help of word-forming means according to word-formation models and in accordance with the rules of word formation have a structure that demonstrates, on the one hand, how their production took place, on the other hand, possible combinations of morphemes available in the language.

This means that derived words have two structures - word-formation and morphemic [21]. The first determines the place of the derived word in the word-formation system and is studied by word-formation proper. The second one demonstrates the types of morphemes, their combinations, meanings, functions and is the object of a section of word-formation science called morphemics. Thus, word formation includes two sections:

1. morphemics;

2. proper word formation.

Word formation proper is combined with morphemics by the unity of synchronous and static approaches. Consequently, morphemics fits into the framework of the system-structural direction. Functional word formation is a part of

functional linguistics, which is addressed to the process of connecting linguistic units of various types in speech activity in order to convey any mental content. The products of speech activity, along with derived words, are simply meanings, borrowings, phrases, sentences, texts.

A small modification of the morphological analysis algorithm can help to recognize misspelled words. To do this, the Levenshtein distance [37] is used – the minimum number of errors, the correction of which leads one word to another. It is believed that there are three types of errors: insertion error, replacement error and omission error. An insertion error is a case when an extra letter is inserted into a word; an omission error is when a letter was omitted; and a replacement error is when one letter is replaced by another. When searching for a word with an error, is need to alternately try to skip letters in the input word (in case of an insertion error), then in the tree (in case of a skip error), or both (in case of a replacement error). At the same time, if skipping a letter in the input string is a trivial action, then skipping a vertex in the tree leads to the fact that it is not known which descendant of this vertex should be selected. In this regard, it is necessary to sort through all the descendants and select all successful options. Errors can manifest themselves in two ways during the course of morphological analysis. When analyzing a word, it reaches a place when there will be no transition along the current letter. An error can be made in one of the previous letters, so the word analysis should start from the beginning [22]. Another way to determine the error in a word is the fact that many paradigms associated with the found prefixes will not overlap with the lexical context must be considered, however, in order to eliminate some of the errors. For example, one of the widespread mistakes is inserting a soft sign into verbs in the third person (in this case, the normal form of the verb is obtained): "жиіжиі" vs " жиі-жиі". A similar word is contained in the dictionary, but it is written with an error that does not allow for syntactic analysis. In addition, correcting errors in a word can give several spellings. To eliminate such situations, the context of the word should be taken into account. For example, "қуаныш тымын" is "қуанышытымын ", or "ктапты" with the correct "кітапты". To correct such errors, syntax analysis with correction or the n-gram method can be used. If an error changes the form of a word, no corresponding combination will be found in their n-gram database. At the same time, it is mandatory to offer replacement options for all words in the n-gram. For

example, the phrase "ата-ана" can also be interpreted as "ата ана". However, in the full phrase "ата-ана", the latter option seems unlikely and will be eliminated. Screening will be carried out either at the stage of homonymy elimination, or at the syntactic analysis, since the phrase does not contain a subject.

2.4 Natural Language Processing methods

Natural Language Processing (NLP) is a topic within machine learning that pertains to recognizing, generating, and processing human verbal and written speech. This research combines artificial intelligence with linguistics. The software engineering profession develops mechanisms for humans and machines to communicate through natural language [11]. A computer is able to understand, read, interpret, and interpret human language, in addition to generating response results as a result of NLP. Generally, processing becomes more efficient when a machine has the ability to decipher human messages into meaningful information 2.1.

Natural language processing algorithms can be used for machine understanding as follows [29]:

- An audio device records the speech of a person.
- Words are converted from audio to written text by the machine.
- Using NLP, a text is divided into its constituent parts, and its context and goal are understood.
- According to the NLP results, the machine decides which commands to execute.

2.4.1 An explanation of language processing

Applications of natural language processing surround us everywhere. These are Yandex and Google searching, translation software, conversational interfaces, artificial intelligence tools like Siri, Alice, Salute from Sber, etc. Many other applications use NLP as well, such as advertising and security. Science and business use NLP technologies: for example, in artificial intelligence research as well as the development of "smart" systems based on natural language processing, from search engines to music software [2].

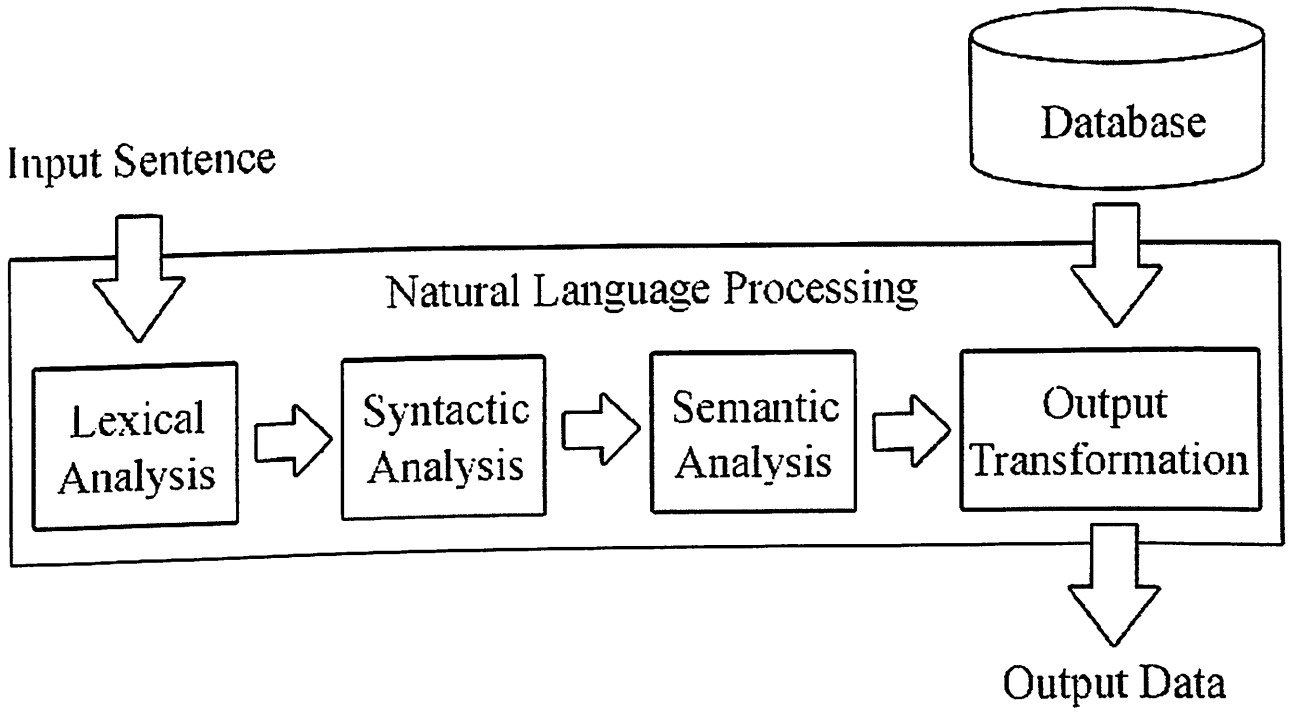


Figure 2.1: NLP Steps

For a long period of time, algorithms were prescribed based on a set of reactions to certain words and phrases, and comparison was used to perform the search. In this case, the machine is not recognizing and understanding the text, but responding to the entered set of characters. An algorithm of this type would not be able to distinguish between a tablespoon and the food served at a school cafeteria [10].

Different approaches to NLP exist. Learning an algorithm involves not just learning words and their meanings, but also learning how sentences are structured, the logic behind it, and how to understand context [3]. A machine must be acquainted with the meanings of the concepts "man" and "suit" in order to understand what "he" refers to in the sentence "a man wore a suit". Specialized algorithms and language analysis methods from basic linguistics are used to teach this to a computer.

As a general term, the field of natural language processing includes a number of subfields. In most cases, these systems use machine learning models, primarily neural networks, and data derived from lots of conversations between individuals.

Because human languages are continuously and spontaneously developing, and for the computer to be able to process the data, it must be clear and structured, certain problems arise during processing, and accuracy is reduced. Furthermore, text analysis methods are heavily influenced by a variety of factors, such as the

language, genre, and topic - more customization is always necessary. Regardless, a large number of natural language processing tasks are still solved by deep learning.

NLP (Natural Language Processing) is an interdisciplinary field that combines computer science and linguistics in order to allow computers to comprehend the language used by humans. The field has become one of the most popular branches of data science. It has been in existence ever since computers were invented.

NLP has been able to reach incredible heights due to the development of technology and computing power. In recent years, technologies that synthesize and recognize speech have become equally popular as technologies that manipulate written texts.

2.4.2 NLP Tasks

Natural language processing enables computers to converse with individuals in their native tongue and to handle various language tasks. NLP, for example, enables computers to read text, hear speech, analyze it, measure emotions, and figure out which sections are significant.

Modern robots are capable of analyzing more linguistic data than humans, without becoming fatigued, and in a constant, impartial manner. Automation will be critical for the proper analysis of text and speech data, given the large volume of unstructured data created every day, ranging from medical records to social networks. Organizing a data source that is very unstructured [14]. The complexity and variety of human languages is astounding. Both vocally and in writing, we express ourselves. Not only are there hundreds of different languages and dialects, but each one has its own set of characteristics.

One of the most promising modern NLP technologies is the use of modern algorithms. NLP is a set of text based and text learning algorithms that allow you to create models with a high level of abstraction in the source text processing architectures containing nonlinear signal transformations. A characteristic feature of NLP algorithms is the passage of incoming information through a much larger number of layers than with traditional (surface) learning. For this algorithm with recurrent architecture, the path that the information signal passes from the input to the output is theoretically unlimited (in practice, it may be limited by the capabilities of the software used) [41]. A large number of familiar and convenient Google services, which are now popular all over the world and in all

fields of application, would be completely impossible without the use of NLP. Consequently, the processing of natural languages through the use of modern algorithms is a promising direction in the theory and practice of sciences of the modern developing information society [1].

Speech recognition This is done by voice assistants of applications and operating systems, "smart" speakers and other similar devices. Speech recognition is also used in chatbots, automatic ordering services, automatic generation of subtitles for videos, voice input, and smart home management. The computer recognizes what the person told him and performs the necessary actions accordingly.

Text processing A person can also communicate with a computer through written text. For example, through the same chatbots and assistants. Some programs work both as voice and text assistants at the same time. An example is assistants in banking applications. In this case, the program processes the received text, recognizes it or classifies it. Then it performs actions based on the data it has received.

Information extraction Specific information can be extracted from text or speech. An example of a task is answering questions in search engines. The algorithm must process an array of input data and select key elements (words) from it, according to which an actual answer to the question will be found. This requires algorithms that are able to distinguish between context and concepts in the text.

Information analysis This is a similar task to the previous one, but the goal is not to get a specific answer, but to analyze the available data according to certain criteria. The machines process the text and determine its emotional coloring, theme, style, genre, etc. This is also true for voice recordings. Information analysis is often used in various types of analytics and in marketing. For example, you can track the average tonality of reviews and statements on a given question. Social networks use such algorithms to search for and block malicious content. In the future, the computer will be able to distinguish fake news from real news, establish

the authorship of the text. NLP is also used when collecting information about a user to display personalized advertising or use information for market analysis.

Text and speech generation The opposite task to recognition is generation, or synthesis. The algorithm must respond to the user's text or speech. This may be an answer to a question, useful information or a funny phrase, but the remark should be on a given topic. In speech recognition systems, sentences are divided into parts. Then, in order to pronounce a certain phrase, the computer saves them, converts them and reproduces them. Of course, distortions can occur at the borders of the "stitching", which is why the voice often sounds unnatural. Text generation is not limited to template responses embedded in the algorithm. Machine learning algorithms are used for it. "Talking" programs can learn from real data. It is possible to ensure that the algorithm writes poems or stories with a logical structure, but they are usually not very meaningful.

Automatic retelling This direction also implies the analysis of information, but both recognition and synthesis are used here.The task is to process a large amount of information and make a brief retelling of it. This can be necessary in business or in science, when it is necessary to obtain the key points of a large data set.

Machine translation Translation programs also use machine learning and NLP algorithms. With their use, the quality of machine translation has increased dramatically, although it still depends on the complexity of the language and is associated with its structural features. The developers strive to make machine translation more accurate and able to give an adequate idea of the meaning of the original in all cases. Machine translation partially automates the task of professional translators: it is used to translate template sections of text, for example, in technical documentation [20].

2.4.3 Text processing

Algorithms do not work with "raw" data: Most of the process is the preparation of text or speech, converting them into a form that can be perceived by a computer.

Clearing Useless data for the machine is removed from the text. These are most punctuation marks, special characters, brackets, tags, etc. Some symbols may be significant in specific cases. For example, in the text about the economy, currency signs carry meaning.

Preprocessing Then comes the big stage of preprocessing - preprocessing. This is bringing information to a form in which it is more understandable to the algorithm.

Popular preprocessing methods:

- Reduction of characters to one case so that all words are written with a small letter;
- Tokenization - splitting text into tokens. This is what individual components are called - words, sentences or phrases;
- Identifying parts of speech for grammar purposes: defining sentences according to parts of speech;
- Lemmatization and stemming - bringing words to a single form. Stemming is rougher, it cuts off suffixes and leaves roots. Words are lemmatized by reducing them to their original word forms, usually taking context into consideration;
- Removal of stop words — articles, interjections, etc.;
- Spell-checking — autocorrection of words that are spelled incorrectly.

The methods are chosen according to the task.

Vectorization After preprocessing, a set of prepared words is obtained at the output. But the algorithms work with numeric data, not with pure text. Therefore, vectors (see Figure 2.2) are created from incoming information — they represent it as a set of numeric values. Popular vectorization options are "bag of words" and "bag of N-grams". In the "bag of words", words are encoded into numbers. Only the number of words in the text is taken into account, not their location and context. N-grams are groups of N words. The algorithm fills the "bag" not with individual words with their frequency, but with groups of several words, and this helps to determine the context.

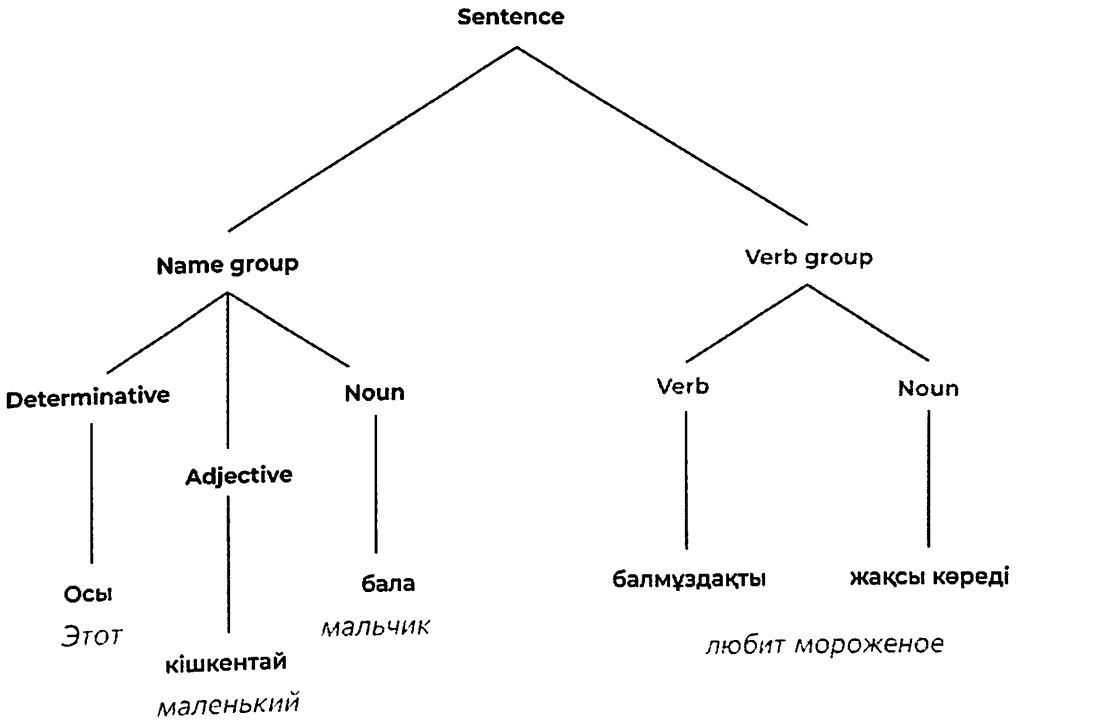


Figure 2.2: Vectorization in NLP

Application of machine learning algorithms Using vectorization, you can estimate how often words occur in the text. But most of the actual tasks are more complicated than just determining the frequency - advanced machine learning algorithms are needed here.

Algorithms process, analyze and recognize input data, and draw conclusions based on them. This is an interesting and complex process, in which there is a lot of mathematics and probability theory.

2.5 Related works in different languages

Since there are a large number of different solutions among automatic spell checking systems (SCS) for the Russian language, a number of criteria have been formed for their selection [38]:

- Possibility of direct access (via the web interface or as a program module);
- The ability to get a list of possible options back (some solutions just underline the words);
- Cross-platform;

- Free or conditionally free access;
- The possibility of word-by-word verification (most solutions are designed forexclusively for checking text corpora).

Since there are a large number of different solutions among automatic spell-checking systems (ASPs) for the Russian language, a number of criteria have been formed for their selection: The following ASPs were selected from a variety of solutions.

Yandex The speller is designed to check and correct spelling errors in Russian, Ukrainian, and English texts. Punctuation, grammatical (word matching errors) and stylistic errors are not corrected. This solution uses the rules of spelling and vocabulary of the modern language, as well as machine learning technology (CatBoost library), which provides correction of words such as "adnikasnii" ("classmates") and takes into account the context ("miss music" – "download music"). In addition, Yandex.The speller uses an algo-rhythm to check off-the-vocabulary words. Language models consist of hundreds of millions of words and phrases. ORTHO dictionary bases are used as dictionaries. This solution uses a web interface with its own API (software interface of the application). This service has a limit on the number of requests: 10 thousand requests per day; and by the number of text volume: 10 million characters per day. For this reason, Yandex can change the volume and number of requests, and also has the right to deny access to the user.

Google Spellchecker Google Spellchecker had its own API until mid-2017. Currently it is part of Google products such as Google Mail, Google Docs, Google Search. This solution is based on machine learning methods, as well as on the rules of spelling and vocabulary of the modern language; the dictionary consists of more than 1 million words. Restriction on requests like Yandex.Speller, 10 thousand requests, no more than 1 million characters, with subsequent billing: $20 for 1 million characters. Spellcheker as Google and Yandex.The speller provides a web interface for processing client requests.

Hunspell Hunspell is a free ASP. It is based on the rules of orthography and vocabulary of the modern language, using MySpell dictionaries, about 200 thousand

words in size, as a dictionary base. In Hunspell, in addition to the error checking function a morphological analyzer is used. This ASP has found its application in solutions such as LibreOffice,OpenOffice, Mozilla Firefox, Thunderbird, Google Chrome, Opera, etc. The main advantages of Hunspell are: support for unicode encoding; use of morphology; application of the n-gram method to determine the correct spelling of words; rules for determining the correct spelling are based on transcriptions; using the C++ programming language. It is worth noting that Hunspell can be used as a tool by the PyEnchant law-checking library, which is implemented in the Python programming language.

LanguageToolis LanguageToolis a shareware ASPP, the price is $5 per month. This solution has support for both web interfaces and software libraries; the Java programming language is used. The size of the dictionary is about 300 thousand words. It is worth noting that the system is focused mainly on English and German. Neural network algorithms are provided for these languages, as well as n-gram language models that are not available for the Russian language. It uses a set of spelling rules and vocabulary of the modern language. Language Tool can be used as an extension for Firefox, Chrome, Google Docs.

Глава 3

Research Method

In this section of the report, the methods of performing the correcting text in morphologically rich languages process will be presented. In order to conduct automatic error correction in Kazakh language, several different approaches have been proposed in the literature review.

An artificial language processing system is understood as a system that performs certain functions by processing speech information in a certain way that is reminiscen of natural language processing. An example might include: creating chatbots to answer user questions, determining the emotional significance of statements, performing machine translations from one language to another, performing language recognition, conducting spelling checks, noting parts of speech in a sentence, and rewriting text information in order to prodIn the context of natural language processing, machine learning is an emerging area of knowledge that is gaining significant attention and has many exciting prospects [33]. great prospects. According to its strictest definition, machine learning can be characterized as a class of artificial intelligence techniques, whose unique feature is not a direct solution to the problem, but the use of a specially trained mathematical model to achieve this result. During the learning process, the model performs a large number of tasks within the desired area.

3.1 Error detection models

At least three methods of automated detection of spelling errors in texts are known:

- statistical
- polygram
- dictionary

With the statistical method, the word forms that make up the text are selected one by one, and their list is ordered according to the frequency of occurrence during the check. Upon completion of viewing the text, an ordered list is presented to the person for control, for example, through the display screen. Orographic errors and typos in any literate text are unsystematic and rare, so that the words distorted by them appear somewhere at the end of the list. Having noticed them here, the controlling person can automatically find them in the text and correct them.

With the polygram method, all two or three letter combinations found in the text (bigrams and trigrams) are checked according to the table of their admissibility in this natural language. If the word form does not contain unacceptable polygrams, it is considered correct, otherwise it is questionable, and then it is presented to the person for visual control and, if necessary, correction.

3.2 Pre-compiled machine dictionary

With the dictionary method, all word forms included in the text, after ordering or without it, in their original text form or after morphological analysis, are compared with the contents of a pre-compiled machine dictionary. If the dictionary allows such a word form, it is considered correct, otherwise it is presented to the controller. He can leave the word as it is; leave it and insert it into the dictionary, so that later in the session such a word will be recognized by the system without comments; replace (correct) the word in this place; require similar substitutions for all further text; edit the word along with its surroundings. Operations on a doubtful part of the text, specified or otherwise possible, can be combined based on the intent of the Automatic correction designer [16].

The results of repeated studies have shown that only the dictionary method saves human labor and minimizes erroneous actions of both kinds - omitting textual errors, on the one hand, and attributing correct words to doubtful ones,

on the other. Therefore, the dictionary method has become dominant, although the polygram method is sometimes used as an auxiliary.

The problem of correcting typos in the names of geographical objects, in contrast to correcting typos in ordinary texts, has its own features that simplify the task. In this case, it is possible to abandon language-specific heuristics and the use of experimental data. In addition, morphological analysis is required to a much lesser extent and syntactic and semantic analysis is not required at all [7]. This is due to the fact that, firstly, the names of geographical objects are presented in a single form (singular, nominative or genitive), there is practically no inflection of the language and, secondly, it makes no sense to resort to determining the interrelationships of the names under study in terms of syntax and semantics. The definition of relationships between the names of geographical objects, for example, that make up a postal address, can only be considered from the point of view of determining the existence of this address. But this is not part of the tasks of the typo correction algorithm. Accordingly, in order to correct typos in the names of geographical objects, a dictionary approach should be applied and it is enough to limit ourselves only to considering the whole name, without dividing it into separate components – the basis, prefixes and affixes. In some cases, you will still have to resort to morphological analysis when the name of a geographical object is represented in the genitive case [36]. From this, a method can be proposed for automatically correcting typos in the names of geographical objects, which consists of the following sequence of steps.

- Step 1: Checking the input value against the reference dictionary
- Step 2: Checking the input value against the reference dictionary by using different variations of the input value
- Step 3: Checking the input value using an additional dictionary with typos
- Step 4: Checking the individual tokens of the composite input value and restoring abbreviations

The input value will mean the name of the geographical object that requires verification, without the type of this geographical object (city, street, etc.). Then the listed steps shown in more detail. Initially, the input value is checked against

the reference dictionary (step 1). If the input value is found in the dictionary, this value is considered correct and does not require further checking for typos. The main step both in terms of complexity and time costs is step 2, which is based on the method of correcting spelling errors by brute force, proposed in the work. Consider a variant of this method for typos like "letter replacement". In the process of searching for a typo, the input value will be modified. The resulting modifications will be called hypotheses of the input value (hereinafter simply hypotheses) [16]. Initially, the input value is fed to the input of the typo correction algorithm. This algorithm is the basis of both step 2 and the entire method of automatic typo correction in general. To apply this algorithm, it is required that the reference dictionary, which will be searched for, be ordered alphabetically.

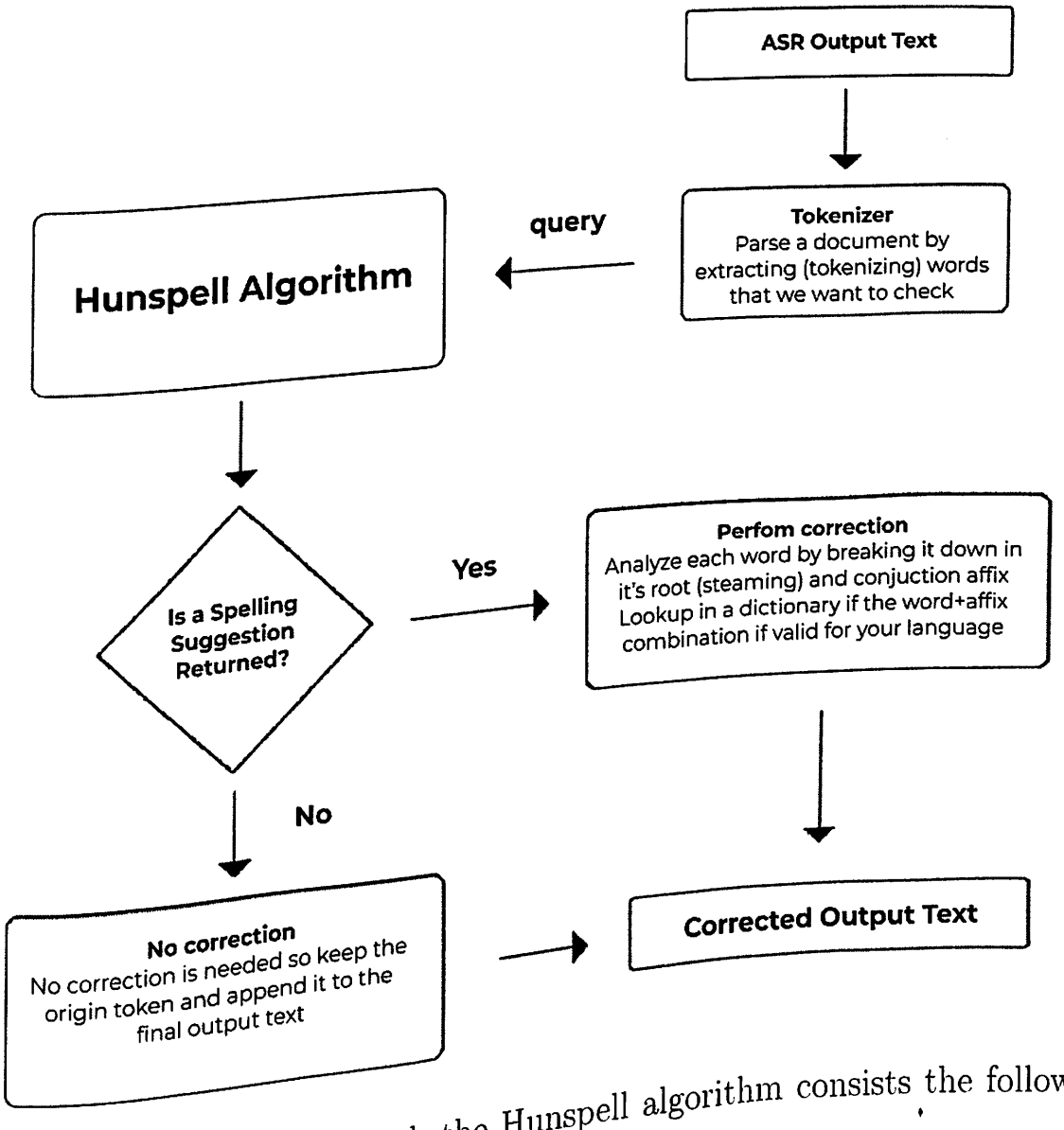


Рис. 3.1: Spell checking text with the Hunspell algorithm consists the following steps

3.3 Automating the correction process

There are three degrees of automation of the text correction process:

1. error detection only
2. detecting them and putting forward hypotheses (alternatives, candidates) for correction
3. detecting errors, putting forward hypotheses and accepting one of them (if at least one is put forward by the system) as an automatically introduced correction

Automatic correction is unthinkable without the first degree. The second and third degrees are possible only with the dictionary method. The second one makes it much easier to make corrections, because in most cases it eliminates the re-selection of a dubious word. The alternatives found are especially useful when the person controlling the text does not know the given natural language or a specific terminological area. However, hypotheses require a lot of searching through the dictionary [27]. Therefore, modern Automatic correctors often have a means of hypothesizing only as an optional, triggered, if necessary, selectively for this dubious word. The third degree of automation is tempting and dangerous at the same time. The temptation lies in the complete automation of the correction process. The danger is that no dictionary, including one enclosed in the human brain, is ever exhaustively complete. When an unfamiliar word is encountered by a system based on an incomplete dictionary, it can "correct"it to the nearest familiar one, sometimes dramatically distorting the original meaning of the text. It is especially dangerous to edit the proper names of persons, companies, products, It is tempting to be able to skip (bypass) proper names and purely special terms, a priori believing them to be correct, but infallible ways of circumventing, especially terms, are not known [28].

3.3.1 Automatic syntactic and semantic analysis

Purely automatic correction could be facilitated by automatic syntactic and semantic analysis of the text being checked, but it has not yet become a part of ordinary Automatic proofreaders. And even if it is available, only a person will be able to

diagnose rapidly changing sets of proper names, terms and abbreviations, as well as occasional verbal innovations that randomly appear. In connection with the above, full automation of corrections can only be applied in any of the following restrictive conditions:

1. The text has the form of a list of terms and terminological phrases in their standard form, so that in Automatic Correction it is enough to have a dictionary that is closed in scope and problematic. At the same time, all the terms are "dissimilar"(for example, there is no "АЛМА"and "АЛЛМА in the dictionary at the same time)

2. Errors are in the nature of replacing the codes of the original letters with the codes of letters that match or are close to the original in outline. For example, ASCII codes of the Russian letters A, B, C, E, Y are replaced by codes of the Latin letters A, B, C, E, Y; Latin letters I and 0 are replaced by numbers I and 0, etc. This also includes repetitions of the same letter that occur due to prolonged pressing of the display key or its malfunction. In the vast majority, if there are more than 2-3 letters in the word form, such corrections are absolutely correct [9].

Ultimately, all methods of semantic analysis are aimed at the formalization and processing of knowledge. However, when solving problems of semantic text processing, a fundamental difference emerges not so much in the methods as in the target setting of knowledge processing: if the logical apparatus is focused on learning and obtaining new knowledge, then semantic text analysis is aimed at extracting ready-made knowledge and their cross-analysis. At the same time, as indicated, the truth or falsity of specific knowledge found in the text is not the goal of semantic processing: the priority goal is to establish semantic relations between lexical elements encoding knowledge. Depending on the degree of formalization of the text and complexity, we can extract from it either the information that it contains – in terms of the components themselves, or other information that reveals the content in terms of contextual knowledge. In the latter case, it is necessary to build a system of contextual knowledge – an ontology [24].

The purpose of semantic processing is to extract ontological meaning from the text and discourse. Thus, the category of "meaning extracted from the mental sphere, acquires certainty and constructiveness, which allows processing by machine

means. On the way to solving fundamental problems, there is the concept of "ontological meaning": it is thanks to it that the essence and structure of this epistemological category acquire a formal form. The ontological meaning is not a mapping of the utterance into a set of zero and one. This is the mapping of an utterance to a coherent set of named concepts, representing a system of knowledge – ontology.

In the most general case, understanding is the goal of communication and the prerogative of two intelligences. At the mental level, the process of understanding means the awakening of reality models (linguistic and figurative) in a person's memory, finding analogies or fixing new connections; at the machine level, the fact of "computer understanding"means the excitation of the corresponding subgraph of meaning on the ontology graph and linking it with other subgraphs. In both cases, a certain linguistic situation (reality) is reconstructed [4]. Combining ontology with meaning allows you to combine the semantics and pragmatics of the text into a single procedural complex. Indeed, the ontology of knowledge is responsible for the connection of the text with non-linguistic reality, and the semantic trajectory is responsible for encoding the ontological meaning in machine memory.

3.4 Hunspell algorithm

Every word in the text d is generated by some theme [31]

$$t \in T \tag{3.1}$$

The text is generated by some distribution on topics

$$p(t \mid d) \tag{3.2}$$

The word is generated by the topic, not by the text

$$p(w \mid d, t) = p(w \mid d) \tag{3.3}$$

In total , such a likelihood function is obtained

$$p(w \mid d) = \sum_{t \in T} p(w \mid t) p(t \mid d) \tag{3.4}$$

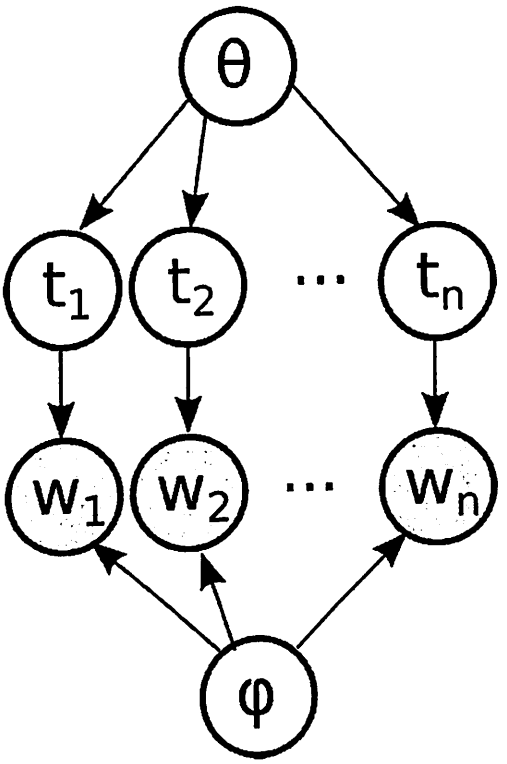


Рис. 3.2: Graphical model of the text

Develop the algorithm evaluation:

$$p(w \mid d) = \frac{n_{wd}}{n_d} \tag{3.5}$$

and needed to find:

$$\phi_{wt} = p(w \mid t) \tag{3.6}$$

$$\theta_{td} = p(t \mid d) \tag{3.7}$$

Maximizing the likelihood

$$p(D) = \prod_{d \in D} \prod_{w \in d} p(d, w)^{n_{dw}} = \prod_{d \in D} \prod_{w \in d} \left[\sum_{t \in T} p(w \mid t) p(t \mid d) \right]^{n_{dw}} \tag{3.8}$$

Maximizing such likelihood by Hunspell algorithm. At the E-step, we are looking for how many words w in the document d from the topic t

$$n_{dwt} = n_{dw} p(t \mid d, w) = n_{dw} \frac{\phi_{wt}\theta_{td}}{\sum_{s \in T} \phi_{ws}\theta_{sd}} \tag{3.9}$$

And at the M-step we recalculate the parameters of the model:

$$\begin{aligned} n_{wt} &= \textstyle\sum_d n_{dwt}, \quad & n_t &= \textstyle\sum_w n_{wt}, \quad & \phi_{wt} &= \frac{n_{wt}}{n_t} \\ n_{td} &= \textstyle\sum_{w \in d} n_{dwt}, \quad & \theta_{td} &= \frac{n_{td}}{n_d}. \end{aligned} \tag{3.10}$$

Parameters that are missing:

- Firstly, such decomposition will not be the only one
- Secondly, there are a lot of parameters, there will obviously be overfitting if the corpus is not orders of magnitude larger than the number of topics
- And it would be absolutely good to get not just a stable solution, but one with some specified good properties

3.4.1 Algorithm usage

The process of work is that a sealed request is sent to the algorithm and all errors that have to be corrected. Usually, mathematically, the problem is posed in this way [26]:

1. the word s is given, passed with errors;
2. there is a dictionary of correct words;
3. for all words w in the dictionary, there are conditional probabilities P(w|s) of what the word w meant, provided that the word s was given;
4. need to find the word w from the dictionary with the maximum probability P(w|s).

This statement — the most elementary — assumes that if a request comes from several words, then the algorithm corrects each word individually. In real

life, the most ideal outcome is to correct the whole phrase, taking into account the compatibility of neighboring words.

There are two unclear points here — where to get a dictionary and how to calculate P(w|s). The first question is considered simple. In 1990, the dictionary was collected from the spell utility database and electronically available dictionaries; in 2009, Google did it easier and simply took the top of the most popular words on the Internet (along with popular misspellings).

The second question is more complicated. If only because its solution usually begins with the application of the Bayes formula!

For now instead of the original incomprehensible probability, it is needed to evaluate two new, slightly more understandable ones: P(s|w) — the probability that when typing the word w, can typo and get s, and P(w) — in principle, the probability of the user using the word w.

3.4.2 Model describing errors

One way or another, it is the need to have some kind of model describing errors and their probabilities.

Such a model is called noisy channel model (noisy channel model) or more briefly error model - error model. This model, to which a separate section is devoted below, will be responsible for accounting for both spelling errors and, in fact, typos [12].

There are different ways to estimate the probability of using the word — P(w). The simplest option is to take for it the frequency with which the word occurs in some large corpus of texts [23]. For an erroneous text that takes into account the context of the phrase, of course, something more complex will be required - another model. This model is called the language model, the language model.

1. A typo occurred:
   - With mistake: Ол жиіжиі басқа қалаларға барады

- Without mistake: Ол жиі-жиі басқа қалаларға барады
- Translation in English: He often goes to other cities

2. Writing in colloquial terms:
   - With mistake: Алма ктапты ктапханадан алды
   - Without mistake: Алма кітапты кітапханадан алды
   - Translation in English: Alma took the book from the library

3. Space errors:
   - With mistake: Сізді көргеніме қуаныш тымын
   - Without mistake: Сізді көргеніме қуаныштымын
   - Translation in English: Glad to see you

4. Hyphen errors:
   - With mistake: Әр ата ана өз баласының бақытты болғанын қалайды
   - Without mistake: Әр ата-ана өз баласының бақытты болғанын қалайды
   - Translation in English: Every parent wants their child to be happy

5. Real-word errors:
   - With mistake: Мен ушақпен ұшпаймын
   - Without mistake: Мен ұшақпен ұшпаймын
   - Translation in English: I don't fly by plane

3.5 Correcting typos

Typos have been made by everyone when composing a search query. Typos can lead to irrelevant or even nonexistent results when mechanisms that correct them are not present. As a result, error correction mechanisms have been built into the search engine to make it more user-friendly [6].

Typos seem to be quite simple to correct at first glance. In that case, implementin a solution may be difficult if it begin with a number of errors. Generally, typos are corrected with two types of context-independent correction and context-dependent correction (taking the dictionary environment into account) [32]. Firstly, the errors are corrected for each word separately, and in a second scenario the error is corrected taking into account the context (for example, for the phrase "Ол уйге барды the error is corrected for each word individually, resulting in "Ол үйге барды while in the second scenario the error is corrected taking into account the context, resulting in "Ол үйге барды").

Only context-independent correction can identify four main categories of errors in the Kazakh-speaking user's search queries:

- typographical errors within the text itself (салем - сәлем - hello), this category includes any kind of omission, insertion, or permutation of letters – 65.7%,
- Combined-separated spellings – 17.4%,
- Layout distort (celtc - сәлем - hello) – 11.2%,
- Verbfigure out (salem - сәлем - hello) – 2,7%,
- Mixed error rate – 9.1%.

Working with a dictionary presented in the form of a tree, the correction occur very quickly and with minimal computational costs [19]. Thus, part of the work with recognition is transferred to work with text, which is computationally much less time-consuming and, as a result, is carried out many times faster. Of course, it is necessary to deal with doubles if we stumble upon them, but the probability of this is small. When recognizing quasi-words, a base with a smaller number of elementary speech units is used. This reduces the learning time of the latter and will play a positive role in creating a multi-vector or dictator-independent database.

Text analysis is the process of obtaining high—quality information from a natural language text. As a rule, statistical training based on templates is used

for this: the input text is separated using templates, then the received data is processed.

Chapter 4

Experiment Setup

In preparation for the publication of the dictionary, scientists from the open source make card files. Kazakh researchers wrote down words on special cards, showing the multiplicity of contexts of using the same word. For this purpose, quotes from a wide variety of literature are selected. As a result, dozens of such cards are attached to each word. Such a process took place in the early 1950s at the Academy of Sciences of the Kazakh SSR. Its employees prepared an "Explanatory dictionary of the Kazakh language". The material (card file) was collected, which reflected the lexical units and the contexts of their use. The total number of records (cards) is 2 million 550 thousand. In them, words presented together with all sorts of meanings from in different contexts. The confusion between the number of linguistic units of the language and the form of their reflection led to the appearance of the described fake. The given facts about how many words there are in the Kazakh language, comparative analysis with other languages of the Turkic group convince of its richness. The development of this living organism continues now, new words are added from year to year.

Thus, a database of words in the Kazakh language can be found in open sources. About 75 thousand correctly spelled words in all types are available in this database. For this experiment, this database was taken from an open source, as well as more than 200 thousand words were parsed from social networks and websites in various formats to achieve a clearer result on training a model for this scientific work (see Table 4.1).

Data base with Kazakh Words	
Definition	Amount of Words
Right written words in all types	74483 words
Parsed words from open source (social media and web-sites) with mistakes	210356 words

Table 4.1: Table with the input data for the experiment

Chapter 5

Results and Discussion

5.1 Results

In this section it will be performed and presented the results of the research work. When there are no candidates available for search, this algorithm works very slowly. It takes the algorithm about a second to process a word of 15 letters, but that performance is doubtful for practical purposes. Performance can be increased by a million times when Hunspell algorithm parameters are added. The way HunSpell works is that for every word from the dictionary, each deletion is added to a separate index, or rather, each word that is obtained by deleting one or more letters (generally 1 and 2) is added to the index. In a similar manner, a deletion is made when a word is searched for candidates, and it is confirmed that it exists in the index. This algorithm handles all possible error conditions - letter substitution, permutations, additions and deletions.

As an example, suppose there is a replacement (in the example, it is taken into account only the first distance). Suppose the initial set includes "алма – apple". In this case, the input is the mistaken word "алмт". There is an index containing the all deletions from "алма", such as: лма, ама, ала, алм. When it comes to "алмт" there is following deletions: лмт, амт, алт, алм. In this case, "алм" is included epy index, so the typographically incorrect "алмт" corresponds to the right written word "alma".

Here is a table below that shows sample results for Kazakh unput data. 200 thousand word samples from social networks and news portals and 75K words from a database of correctly written Kazakh words were used for training as prescribed in the experiment. The initial sample was divided into 2 parts, 90%

Results						
Input data	The percent of Errors	Errors in Top 7	Accuracy	Accuracy in Top 7	Weak	Effort (words per second)
Hunspell Algori thm	11,78%	14,82%	52,24%	71.98%	6,93%	164%

Table 5.1: Table with the input data for the experiment

Results						
Input data	The percent of Errors	Errors in Top 7	Accuracy	Accuracy in Top 7	Weak	Effort (words per second)
Hunspell Algori thm	8,94%	10,76%	38,29%	69.34%	2,65%	279%

Table 5.2: Table with the input data for the experiment

was used for training, 10% for evaluation. Results (see Table 5.1):

In order for the experiment to be as pure as possible, a check was made on a voluminous literary text. Text metrics from the work "Абайдың қара сөздері" - «Abai Book of Words» (see Table 5.2).

To begin with, the text can be viewed from two points of view: either as a simple sequence of words, spaces and punctuation marks, or as a network of interconnected syntactic-semantic dependencies of concepts. For example, in the sentence "I love big dogs", it can be arranged the words in any order, while the structure of the links between the words will be the same: Errors can also occur at different levels. A mistake is made in the linear structure — type the same word twice, forget the period, parenthesis, and similar things. That is, an error occurs in the process of building words into a chain, and a person is unlikely to commit it due to ignorance of any grammatical rule. Probably, taking into account about simple inattention. However, at the chain level, some real grammatical errors can also be identified. For example, in the Kazakh text there

should be no combinations of "кеше бара жатырмын" — there should be either "кеше бардым" or "бүгін бара жатырмын".

But still, the real grammar begins at the level of the analysis of the network of concepts. For example, in our example about dogs, the subject "I" should be combined in person and number with the verb "love" regardless of the relative position of these words in the sentence. Of course, this consideration does not negate the need to catch simpler errors identified at the level of the linear structure.

When a word with a typo is found, the word of this word is divided into trigrams and a search is performed for words from the dictionary containing trigrams. The search occurs only for words in which the character difference is no more than 3. Words in the dictionary with the highest trigram content become candidates for replacing a word with a typo.

When using the method of searching for trigrams in the word, consisting of five letters and having a typo in the third letter, it is not possible to find the right candidate, since in any search from the dictionary there will not be the necessary word that includes at least one of the trigrams in the word with a typo.

5.2 Discussion

Applied morphological analysis and the definition of a part of writing in Kazakh are fairly laborious. This language is one of the most difficult Turkic dialects. Each word in the language can have a vast array of word forms formed by adding suffixes and endings to it. One of the most popular dictionaries for the study of Kazakh in the world is the Morphological Dictionary of the Kazakh Language. However, there is no comprehensive database of Kazakh words at present. In relation to the development of computational linguistics in Kazakhstan in general, the development of a morphological Kazakh dictionary is a priority. It is arranged alphabetically by morphemes, i.e., according to the composition, dividing the words into significant parts. The Dictionary can be used by teachers and students to check their results when the morphemic analysis is difficult.

5.2.1 Algorithm Optimization

It was necessary to develop a method of evaluating the quality of the spellchecker before writing it directly. Using a typo collection that Hunspell had created, Hun-

spell presented misspelled words along with their correct counterparts. This case, however, prevents the use of this method due to the lack of context. The problem was solved by writing a simple typo generator first. An input word is fed to the typo generator, which outputs a word with a specified number of errors. These errors are composed of rearranging two letters in a particular order, replacing one letter with another, deleting one, and inserting a new letter. It is possible to define the probability of each type of error separately, and the probability of making two typos (which depends on the word length) is also calculated.

5.2.2 Accuracy

Verifiers of spelling and grammar are implemented either as standalone applications or as part of other applications, like word processors, email clients, or even search engines. At the same time, this system has a mandatory set of common functions, consisting of the search for errors, the highlighting of errors in the text (either through colouring or underlining) and the display of options for correctVerification requests are transmitted to the system with source text that needs to be verified.itted to the system. Several words or entire paragraphs may be included in the text. The parsing process further divides the text into individual words or into whole phrases, based on the algorithm used (the main words together with their particles).

5.2.3 Resource need

Spell-checking automation tools include spell-checking and grammar-checking tools. The word processor allows you to implement two spell checking modes automatic and command.

The built-in automatic spell checker is essentially an expert system and allows customization. The built-in dictionary of the spell check system is not editable. All additions and changes are made to a special plug-in user dictionary. Reviewing can be understood as two processes: editing the text with registration of changes and commenting on the text. Unlike the usual editing, when reviewing the text of the document, it does not change completely — the new version and the old one "coexist" within the same document on the rights of different versions. The absence of grammatical errors is one of the main characteristics of the text.

Grammar errors can occur due to a number of factors. First, a person may be ignorant of the rules of grammar, and second, a typo may occur when typing. It is built into the developed environment that an automated spell-checking system is available to prevent grammatical errors. There is a database of spelling variants of Kazakh words, as well as knowledge of grammar rules, which constitute the basis of this system. In addition to comparing each written word to the database, the system also checks the consistency of spelling (spelling consistency, comma placement, etc.). Errors are indicated with hints and sometimes correction methods are given. A natural language text processing system such as this one can be used as an example. By default, the algorithm detects and eliminates any spelling or grammatical errors automatically.

Chapter 6

Conclusion

The subject of the research of this scientific work is the correction of words and text, namely editing, processing and presentation of the entered text with errors in the correct grammatical and syntactic form. Based on this study, a linguistically unbiased model for correcting spelling errors is presented and applied to Kazakh language. A new algorithm is developed that outperforms the old leading system. The system also has the merit of allowing arbitrary features at both the word level and sentence level to be incorporated. The main determinant of the effectiveness of spelling correctors was observed to be the quality of the language model during experiments with features of varying types.

In addition, morphological and semantic information can also be helpful. There are three steps in the direction of future work: the first is to augment traditional language models with neural ones and determine if this can better handle long-distance dependencies, which could be helpful in selecting the best candidate. The second step is to apply this developed model to languages with complex morphology and see if the same features are beneficial as in Kazakh's case. Then the third method is to use finite-state tools to re-implement our model since the main components (candidate search and candidate ranking) are indeed finite-state operations.

The ready-made software solution is based on carefully compiled and proven algorithms for analysis and synthesis, and has the following functionality:

- text recognition
- checking and correcting spelling, syntactic, punctuation errors

- correcting signs according to uniform accepted standards (dashes, hyphens, quotes

Most of the functionality is implemented by connecting a large database of dictionaries. The peculiarity of the automatic text correction algorithm is due to the fact that the ready-made software solution is able to correctly process text not only for the presence of spelling and punctuation errors, which in itself is common for text analyzers. The algorithm can also detect errors from a grammatical point of view. And this already requires analyzing the text from a semantic point of view. The parser recognizes the structure of a sentence, namely the syntactic dependencies of words. As a result, a syntactic tree is built and the components are identified. Usually, the grammar is built so that the output is a syntactic tree that allows performing various transformations of lexical content with the re-agreement of dependent words, and it is also easy to distinguish semantics, in particular, to use an algorithm for weighing alternative options for building a tree. The analyzed sentences may have different complexity, include unknown words or deviations from the normative syntax. To effectively cope with different tasks, the parser uses several different algorithms, including structural top-down analysis and bottom-up analysis, and also applies semantic analysis to refine the results in case of ambiguities. Multilingual translation models will emerge in the near future, along with transfer learning and substantially larger monocorpora that will complement parallel corpora. As a result, translation quality will be significantly improved for languages with low resources (with comparatively small training samples). As machine translation becomes more proficient at understanding context and subject matter, manual translation will be completely replaced by machine translation. There is a possibility that machine simultaneous interpretation will emerge in the future five to ten years. For more accurate work, an algorithm has been developed that combines the advantages of ascending and descending probabilistic parsing of a sentence. Top-down syntactic analysis, or analysis through synthesis, begins by making assumptions about the large-scale structure of a sentence, and then clarifies and details this assumption, recursively descending to the level of specific words. In other words, this algorithm initiates parsing from the initial nonterminal S. Bottom-up syntactic analysis begins parsing with specific words, first linking pairs of words, then connecting new words or other related pairs to these pairs. Gradually, the linking process reaches the

initial nonterminal S - that is, all the words in the sentence are connected into a single structure. The developed automatic text corrector is the most accurate, since the algorithm uses a large database of Kazakh words, the algorithm of text syntactic analysis is optimized. The algorithm takes into account and parses not only the semantic part, but also the sign part (removes double spaces, uses dashes, hyphens correctly, places quotes uniformly, etc.)

References

[1] S Amit, Kumar Shukla Rajendra, and Ajit Paul. "Spelling Error Assessment Module for Answer Evaluation Process". In: *Communications on Applied Electronics* 7 (2017), pp. 23–26.

[2] Charles R. Blair. "A Program for Correcting Spelling Errors". In: *Inf. Control.* 3 (1960), pp. 60–67.

[3] I. A. Bol'shakov. "Automatic error correction in inflected languages". In: *Journal of Soviet Mathematics* 56 (1991), pp. 2263–2279.

[4] Dustin Boswell. "CSE 256 (Spring 2004) "Language Models for Spelling Correction"". In: 2004.

[5] Eric Brill and Robert C. Moore. "An Improved Error Model for Noisy Channel Spelling Correction". In: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong: Association for Computational Linguistics, Oct. 2000, pp. 286–293. DOI: 10.3115/1075218.1075255. URL: https://aclanthology.org/P00-1037.

[6] Noam Chomsky. "Syntactic structures". In: *Syntactic Structures*. De Gruyter Mouton, 2009.

[7] Kenneth Ward Church and William A. Gale. "Probability scoring for spelling correction". In: *Statistics and Computing* 1 (1991), pp. 93–103.

[8] Fred J. Damerau. "A Technique for Computer Detection and Correction of Spelling Errors". In: *Commun. ACM* 7.3 (Mar. 1964), pp. 171–176. ISSN: 0001-0782. DOI: 10.1145/363958.363994. URL: https://doi.org/10.1145/363958.363994.

[9] Fred J. Damerau. "A technique for computer detection and correction of spelling errors". In: *Commun. ACM* 7 (1964), pp. 171–176.

[10] Sebastian van Delden, D. Bracewell, and Fernando Gomez. "Supervised and unsupervised automatic spelling correction algorithms". In: *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004.* (2004), pp. 530–535.

[11] S Haas. *Tools for Natural language processing.* 2011.

[12] Allen R. Hanson, Edward M. Riseman, and E. Fisher. "Context in word recognition". In: *Pattern Recognit.* 8 (1976), pp. 35–45.

[13] Graeme Hirst and Alexander Budanitsky. "Budanitsky, A.: Correcting real-word spelling errors by restoring lexical cohesion. Nat. Lang. Eng. 11(1), 87-111". In: *Natural Language Engineering* 11 (Mar. 2005), pp. 87–111. DOI: 10.1017/S1351324904003560.

[14] Neil Houlsby et al. "Parameter-Efficient Transfer Learning for NLP". In: *ICML.* 2019.

[15] John HUTCHINS. "The first public demonstration of machine translation : the Georgetown-IBM system, 7th January 1954". In: *http://www.hutchinsweb.* *IBM-2005.pdf* (). URL: https://cir.nii.ac.jp/crid/1573105976172429

[16] Dan Jurafsky and James H. Martin. "Speech and language processing - an introduction to natural language processing, computational linguistics, and speech recognition". In: *Prentice Hall series in artificial intelligence.* 2000.

[17] Zhanibek Kozhirbayev and Zhandos Yessenbayev. "Kazakh Text Normalization using Machine Translation Approaches". English. In: *CEUR Workshop Proceedings* 2780 (2020). ISSN: 1613-0073.

[18] Adam Lally and Paul Fodor. "Natural language processing with prolog in the ibm watson system". In: *The Association for Logic Programming (ALP) Newsletter* 9 (2011).

[19] Diane Larsen-Freeman. "On the need for a new understanding of language and its development". In: *Journal of Applied Linguistics and Professional Practice* 3.3 (Sept. 2015), pp. 281–304. DOI: 10.1558/japl.v3.i3.281. URL: https://journal.equinoxpub.com/JALPP/article/view/13207.

[20] Jun Lin and Johannes Ledolter. "A Simple and Practical Approach to Improve Misspellings in OCR Text". In: *ArXiv* abs/2106.12030 (2021).

[21] Aibek Makazhanov et al. "Spelling Correction for Kazakh". In: Apr. 2014. ISBN: 978-3-642-54902-1. DOI: 10.1007/978-3-642-54903-8_44.

[22] Aibek Makazhanov et al. "Spelling correction for kazakh". English. In: *Computational Linguistics and Intelligent Text Processing - 15th International Conference, CICLing 2014, Proceedings*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) PART 2. 15th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing 2014 ; Conference date: 06-04-2014 Through 12-04-2014. Germany: Springer Verlag, 2014, pp. 533–541. ISBN: 9783642549021. DOI: 10.1007/978-3-642-54903-8_44.

[23] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[24] Eric Mays, Fred J. Damerau, and Robert L. Mercer. "Context based spelling correction". In: *Inf. Process. Manag.* 27 (1991), pp. 517–522.

[25] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. "Natural language processing: an introduction". In: *Journal of the American Medical Informatics Association* 18.5 (Sept. 2011), pp. 544–551. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2011-000464. eprint: https://academic.oup.com/jamia/article-pdf/18/5/544/5962687/18-5-544.pdf. URL: https://doi.org/10.1136/amiajnl-2011-000464.

[26] Mohammed Nejja and Abdellah Yousfi. "The Context in Automatic Spell Correction". In: *Procedia Computer Science* 73 (2015), pp. 109–114.

[27] Hwee Tou Ng et al. "The CoNLL-2013 Shared Task on Grammatical Error Correction". In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1–12. URL: https://aclanthology.org/W13-3601.

[28] Franz Josef Och. "Minimum Error Rate Training in Statistical Machine Translation". In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, July 2003, pp. 160–167. DOI: 10.3115/1075096.1075117. URL: https://aclanthology.org/P03-1021.

[29] T Pedersen. *Free NLP Software*. 2011.

[30] Tommi Pirinen and Krister Lindén. "Creating and Weighting Hunspell Dictionaries as Finite-State Automata". In: *Investigationes Linguisticae* 21 (June 2010). DOI: 10.14746/il.2010.21.1.

[31] Tommi Pirinen and Krister Lindén. "Creating and Weighting Hunspell Dictionaries as Finite-State Automata". In: *Investigationes Linguisticae* 21 (June 2010). DOI: 10.14746/il.2010.21.1.

[32] Tommi A. Pirinen and Krister Lindén. "State-of-the-Art in Weighted Finite-State Spell-Checking". In: *Computational Linguistics and Intelligent Text Processing*. Ed. by Alexander Gelbukh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 519–532. ISBN: 978-3-642-54903-8.

[33] Joseph J. Pollock and Antonio Zamora. "Automatic spelling correction in scientific and scholarly text". In: *Commun. ACM* 27 (1984), pp. 358–368.

[34] Joseph J. Pollock and Antonio Zamora. "Collection and characterization of spelling errors in scientific and scholarly text". In: *J. Am. Soc. Inf. Sci.* 34 (1983), pp. 51–58.

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

[36] Gaoqi Rao et al. "Overview of NLPTEA-2018 Share Task Chinese Grammatical Error Diagnosis". In: *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 42–51. DOI: 10.18653/v1/W18-3706. URL: https://aclanthology.org/W18-3706.

[37] Alla Rozovskaya and Dan Roth. "Grammar Error Correction in Morphologically Rich Languages: The Case of Russian". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 1–17. DOI: 10.1162/tacl_a_00251. URL: https://açlanthology.org/Q19-1001.

[38] Alexey Sorokin. "Spelling Correction for Morphologically Rich Language: a Case Study of Russian". In: Jan. 2017, pp. 45–53. DOI: 10.18653/v1/W17-1408.

[39] Kristina Toutanova and Robert C. Moore. "Pronunciation Modeling for Improved Spelling Correction". In: *ACL*. 2002.

[40] L. Amber Wilcox-O'Hearn, Graeme Hirst, and Alexander Budanitsky. "Real-Word Spelling Correction with Trigrams: A Reconsideration of the Mays, Damerau, and Mercer Model". In: *CICLing*. 2008.

[41] Emmanuel J. Yannakoudakis and David Fawthrop. "The rules of spelling errors". In: *Inf. Process. Manag.* 19 (1983), pp. 87–99.