

# CLASSIFICATION MODEL AND MORPHOLOGICAL ANALYSIS IN MULTILINGUAL SCIENTIFIC AND EDUCATIONAL INFORMATION SYSTEMS

FEDOTOV A.M.<sup>1</sup>, TUSSUPOV J.A.<sup>2</sup>, SAMBETBAYEVA M.A.<sup>2</sup>, FEDOTOVA O.A.<sup>3</sup>,  
SAGNAYEVA S.K.<sup>2</sup>, BAPANOV A.A.<sup>2</sup>, TAZHIBAIEVA S.ZH.<sup>2</sup>

<sup>1</sup>Institute of Computational Technologies of the Siberian Branch of the RAS, Novosibirsk, Russia

<sup>2</sup>L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

<sup>3</sup>State Public Scientific and Technological Library of the Siberian Branch of the RAS, Novosibirsk, Russia

E-mail: aigerian@mail.ru, sagnaeva\_tar@mail.ru, madina\_jgtu@mail.ru, fedotov@ict.nsc.ru,  
tussupov@mail.ru

## ABSTRACT

The article describes the issues of building models of documentary and factographic search in multilingual scientific and educational information systems, working with documents of rather free structure. A model of information system document classification is proposed based on the use of tolerance relation, taking into account possible absence of a priori defined classifiers. Particular attention is paid to formation of feature space, taking into account the morphology of the document language. The article contains an overview of morphological text analyzers that can be used to determine the normal form of word. The features of application of morphological analyzers are described; their advantages and disadvantages are listed.

The rules for normalization of words of the Kazakh and the algorithm to handle both vocabulary and non-vocabulary (including non-existent) words are developed. A multilingual thesaurus of scientific and technical concepts (terms) on information technology in English, Russian and Kazakh is developed. The system of term normalization and interlingual compliance is implemented for it.

**Keywords:** *factographic search, educational information systems, document classification, tolerance relation, morphology, morphological text analyzers.*

## 1. INTRODUCTION

In the process of scientific and especially educational activity, plenty of time and energy is consumed by the work with various materials, documents and references given in the text form: search for required documents, systematization and classification of documents in accordance with the task set. To meet the information needs of modern users, a support for complex information search or sorting (classification)<sup>1</sup> functions is needed, as well as an option to view resources by categories (headings) and vocabularies-classifiers.

The concept of information search was first introduced by the American scientist Calvin

Mooers<sup>2</sup>, who noticed that the search is carried out in order to meet the information need of the user expressed in the form of information request. Information request is a formalized statement of a natural language. The objects of search are the documents, information about them, facts, data and knowledge that best meets the request (relevancy), and information needs of the user (pertinence). An important place in this science deals with issues of direct meeting the information needs of the user. Systematization of Mooers results and their generalization was described in the monograph of the employees of the All-Union Institute for Scientific and Technical Information (VINITI) [2].

<sup>1</sup> The tasks of information retrieval and sorting are algorithmically equivalent, see [1].

<sup>2</sup> Calvin Northrup Mooers (1919 – 1994) – the founder of the scientific approach to information retrieval, in 1950 introduced the terms of “information retrieval”, “information retrieval system”, “information retrieval language”, “search image”, “descriptor”, “descriptor dictionary” and others.



Another monograph [3] published by VINITI and dedicated to methodological foundations of theoretical computer science, provides a detailed overview of the theoretical problems of information and factographic search based on identification of two types of information needs of a user: the need for information about the sources of necessary scientific information and the need for necessary scientific information itself – it is said that to meet information needs of the former, information systems called “documentary” are intended, and for the latter – factographic systems are intended. Currently, the most popular means of information support of scientific and educational activities are intelligent systems (IS), combining the capabilities of information systems of both of the above types and allowing to satisfy information needs of a skilled user in accordance with ‘document – fact – reasoning’ scheme [4, 5]. In the first place, such systems search for information resources (documents), which may contain the needed information. Then bibliographic information about the source of information is determined. An important step in the process of functioning of such systems is extraction of the facts contained in document texts, that is, in the most general sense, “a special kind of sentences securing empirical knowledge”, corresponding to the information request [6].

An additional feature of scientific and educational information systems for countries such as Russia or Kazakhstan is the need to support the search and classification processes simultaneously in several languages: for Russia – mainly in two languages (Russian and English), and for Kazakhstan – minimum three (Russian, English, and Kazakh). Thus, the documents must be indexed in three different feature spaces with equivalence relations between their elements.

Note that elements of feature spaces can be given in different word forms in the document, so the most important problem is the account of the morphology of a particular language in document indexing. English and Russian refer to a group of inflected languages, which are characterized by a developed system of inflection. Kazakh is a Turkic language of Kipchak group, which is a type of synthetic agglutinative languages<sup>3</sup> having a rich and complex morphology [7, 8]. Its words are usually composed of a stem and affixes

<sup>3</sup>Agglutinative language (from the Latin word Agglutinatio – “sticking”) is a language that has a system where the dominant type of inflection is agglutination (“sticking”) of different prefixes or suffixes, and each of them has only one meaning.

(suffix+ending) added to it (usually at least two or three).

Different languages have different semantic and grammatical features, so the algorithms often successfully used to process one language, show a very low efficiency in another language. Note that accounting all word forms for each word in Kazakh is by an order of magnitude greater in terms of complexity of text processing.

The article provides a comparative analysis of morphology accounting models (with a dictionary and without a dictionary) for English and Russian. A particular attention is paid to the development of the morphological model of the Kazakh language.

## 2. DOCUMENT CLASSIFICATION MODEL

Since the tasks of search and classification of information are equivalent [1], it is sufficient to consider document classification model that most adequately reflects the peculiarities of working with information systems intended to support research and educational activities [9].

The most common way of classifying documents is a facet classification. Its formation theory is formalized by the Indian library scientist S.R. Ranganathan (see [10]). The objects are classified simultaneously by several features (facets) independent from each other. As applied to digital documents (and generally electronic resources), metadata elements act as the facets. Metadata structure is described in detail in [9]. Key terms represent a special kind of metadata. Key terms (descriptors) are the basic meaning content of the text, which are expressed by the list of full words selected from either the text itself or from its title, or from a special standard dictionary (thesaurus) [2, 4]. Note that focusing on a thesaurus rather greatly simplifies the problem considered. We have considered the problem of information classification in a favorite, fairly narrow subject area of interest to the user, dedicated to computer science.

It is important to note that in development of scientific and educational information systems for which bibliographic features of the documents are far less important compared with conventional features, subsets of bibliographic metadata values that form the facet value, are usually more wide. Thus, the links to various reprints (stereotypical reissues) of the same document in terms of scientific and educational systems is advisable to be considered equivalent.

Describe a simple formal model of document classification using metadata (see [9, 11]). Given that the information system architecture is multi-level [9] and includes at a minimum the following components: data storage – repository, metadata server, application server and server supporting work with regulatory dictionaries (dictionary-reference book), the main load of indexing and describing the documents falls on metadata server.

Metadata server stores a catalog of the information system, where any document  $d_i$  circulating in the information system is represented as  $d_i = \{m_i^{j,k}\}$ , where  $m_i^{j,k}$  are values of metadata elements  $M^j$ ,  $k$  is the number of values (including repetitions) of the corresponding metadata element in the document description. Consider a subset of metadata values  $M_C$ , defining a set of classification features of documents used to make search prescription (taking into account logical operations specified). For a fixed metadata element  $M^j$ , where  $M^j \in M_C$ , subsets  $M_i^j$  of the set of values of this metadata element are pre-determined (generally speaking, specified subsets may overlap).

We will consider two documents tolerant (recall that tolerance is a relation that has the properties of reflexivity and symmetry, but generally speaking, may not have transitivity property in contrast to equivalence relation; the properties of this relation are studied in detail in [12]), if their values of a certain metadata element are included in the same subset  $M_i^j$ , while if the values of the metadata element considered may be repeated, then the documents are considered tolerant at coincidence of at least one of the values. Each of these subsets gives rise to sub-class of tolerance in the set of documents, let's denote it as  $K_i^j$ .

Moreover, in most cases such sub-classes are maximal, i.e. are classes of tolerance. Sub-class  $K_n^j$  is the class if there is no other sub-class  $K_l^j$  differing from it (i.e. generated by another set of metadata elements), such that  $K_n^j \subset K_l^j$ , otherwise  $K_n^j$  is not a class.

Let's clarify when sub-classes are not classes (this is necessary, for example, to determine tolerance space basis described below). First of all, if  $M_i^j \subset M_n^j$ , then  $K_n^j \subset K_i^j$ , and thus  $K_n^j$  is not a class, except for the particular selection of documents when  $K_n^j = K_i^j$ , but in this case, it also obviously makes no sense to consider  $K_n^j$  as a separate class. From a substantive point of view,

this situation corresponds to the inclusion of a certain classifier section to a section of a higher level, when both these sections are taken into account when describing the tolerance space (of course, we may not take into a lower level when determining tolerant elements, but then we will have to deal with tolerance space different from the initial one). In this situation, sub-classes, not being classes, are defined a priori.

However, the situation is possible when  $K_n^j \subset K_l^j$  due to specific features of documents. For example, in the information system of the history of mathematics, all documents having geographical feature "Egypt", have chronological feature "BC". And the documents relating to other regions also have the said chronological features. It is clear that in this case all the documents with "Egypt" feature are mutually tolerant not only due to geographical, but also due to chronological feature, however, the emergence of at least one document with "Egypt" feature dating to a new era, will change this situation. Thus, in this situation, it is appropriate to consider sub-class  $K_n^j$  (for example, when building the basis) as a class.

The set of all tolerance classes (including sub-classes considered in accordance with the above as a class) will be denoted as  $H$ .

Further, describe the structure of a basis of the described tolerance space (some set  $H_B$  of tolerance classes is called basis if for every pair of tolerant documents there is a class of  $H_B$ , containing both these documents and removal of at least one class from  $H_B$  leads to loss of this property). Obviously, the set of tolerance classes  $H_M$  (in our structure also including sub-classes considered as classes), generated by the set of subsets  $M_i^j$ , contains a basis. We cannot say that  $H_M$  is exactly the basis because sub-classed included in it are not classes, and they can be removed without the loss of the first property from the basis definition. However, since the addition of even one document to the system database can make a sub-class a class and, therefore – a "fully-featured" basis element, then consideration of such sub-classes as the elements of the basis is reasonable in terms of organization of classification and search for documents.

Description of tolerance classes is of great practical importance. First, consider the set of all documents for which there exists a set of classes (including sub-classes considered as classes) of  $H$ ,

that each of these documents is included in these and only these classes. This set is a core of tolerance, and the set of all cores of tolerance defines an equivalence relation on the set of documents. In addition, to build cores of tolerance, it is sufficient to consider only the classes (and sub-classes) from basis  $H_M$  [12].

Thus, search representation containing a subset of metadata that defines a set of classification features indicating the combination of the values of these metadata using logic operations, determines a certain core of tolerance on the set of documents that is issued to the user as a response to its information request.

In addition, on the set of tolerance classes, we can also in turn introduce the tolerance relation. Wherein the classes are considered tolerant having at least one common document. This structure is useful, for example, to organize search for documents “by analogy”.

The formalism based on the use of tolerance relation turns out to be more convenient when creating digital libraries, as opposed to conventional libraries, where classifiers are specified a priori. When working with digital libraries, you often have to use particular document clustering algorithms (see, for example [4]), and only after that, based on clustering results, define subsets of the sets of values of metadata elements, acting as facet values.

Thus, the search image of the document is a set of metadata values  $d_i = \{m_i^{j,k}\}$ , which can be considered as a vector of some feature space. The process of determining a set of metadata values in the classical literature on information retrieval is called “coordinate indexing” (see, for example, [13]).

The set of metadata values is divided into two disjoint subsets unequal by weight:

$$M = M_S \cup M_T,$$

where  $M_S$  is a set of values of system metadata such as bibliographic description, list of authors, year of publication, publisher, title, etc.,  $M_T$  is a set of key terms belonging to a thesaurus (a list of descriptors). In turn, the set  $M_T$  consists of  $M_{AT}$  – author’s key terms,  $M_{DT}$  – key terms describing the document and  $M_{ST}$  – key terms found in the document text and system metadata.

In automation of the process of coordinate indexing, which is to automate the process of making the set  $M_{ST}$ , the problem arises of

identifying words of text in a natural language, bringing them to the normalized form and their matching with the key terms from the thesaurus. It is solved using a morphological analyzer defining morphological features of words of text and normalized word forms.

### 3. REVIEW OF THE EXISTING SOLUTIONS TO MORPHOLOGICAL ANALYSIS

In linguistics, morphological analysis is defined as a procedure resulting in retrieval of information about word’s internal structure from the form of its external appearance. *Morphological analysis* provides determination of the normal form used to build a given word form, and a set of parameters, assigned to this word form. Morphological analysis may be performed: a) by division of word form into the stem and supposed ending with their subsequent verification of compatibility; b) by final combinations of letters; c) by using universal mathematical models of morphology allowing to normalize word forms by calculations.

There are several dozen algorithms of morphological analysis for different languages. The basis of construction of algorithms for morphological analysis is division of all words into classes that define the behavior of the letter structure of word forms. These classes are called morphological. Changes in word forms may be of a different nature and may be associated both with changes in word stem, and with changes in its ending. In the first case a morphological class is called stem-changing, in the second case – inflected.

There are the following types of morphological analysis: with a dictionary of word forms, with a dictionary of the stems, by logical multiplication and without a dictionary. In case of morphological analysis with a dictionary of word forms, there is no division of words. Such dictionary shall almost cover all possible word forms of the language, and if the word is not found in the dictionary, it shall try to determine its part of speech to use for syntactic analysis.

In case of morphological analysis with a dictionary of the stems, the dictionary of the stems is used the stems of simple and complex words without internal inflection, and auxiliary tables containing a list of endings arranged in accordance with grammatical forms (such as gender, number, case). If a word has several forms of the stems, all of them are included in the dictionary. Each stem

is assigned a combination of code of connective class and code of inflectional class, and if the stem is homonymous with others – it is assigned a series of combinations of such codes.

The method of logical multiplication involves the use of functions defined on word forms and comparing each word form with some information. Dictionary function is a function defined on word forms and comparing each word form with some information represented as a logical conjunction of morphemes included in this word form. This method is applicable to inflected languages and suggests the existence of a dictionary of the stems.

In case of the use of dictionaries, the drawback is inability to create a full dictionary for the subject area, even though they provide the most comprehensive analysis of word form.

Morphological analysis without a dictionary is performed without referring to a dictionary, just by

using tables of affixes and a special list of words with no grammatical meaning. Algorithms of programs operating without a dictionary use probabilistic and statistical methods and lexicons of (quasi-) suffixes, (quasi-) stems constructed empirically. The drawback of no-dictionary approach is primarily that probabilistic and statistical methods do not work well with a small sample, a large volume of lexicons is generated using the method, and the accuracy of such analysis is much lower than for systems operating with a dictionary.

To date, a variety of software tools are developed for morphological analysis of all words of text (translation of a word to a normal form, generation of morphological features for them). Comparative description of morphological analyzers (with a dictionary and without a dictionary) for languages of inflection group is given in Table 1.

Table 1 – Morphological Analysis Software

| Name              | Methods           | License                     | Platform                     | Console | API | Modularity  | Languages     |
|-------------------|-------------------|-----------------------------|------------------------------|---------|-----|-------------|---------------|
| Shareware         |                   |                             |                              |         |     |             |               |
| AOT               | dictionary        | LGPL <sup>4</sup>           | GNU/Linux, Microsoft Windows | -       | +   |             | RUS, ENG, DEU |
| MAnalyzer         | dictionary        | MIT <sup>5</sup>            | GNU/Linux                    | -       | -   | Library     | RUS, ENG      |
| Myaso             | Viterbi algorithm | MIT                         | Ruby                         | -       | +   | Library     | RUS, ENG,     |
| Mystem            | dictionary        | Noncommercial               | GNU/Linux, Microsoft Windows | +       | +   |             | RUS           |
| Phpmorphy         | dictionary        | LGPL                        | PHP                          | -       | +   | Library     | RUS, ENG, DEU |
| Pullenti SDK      | no data           | Shareware                   | .NET                         | -       | +   | SDK module  | RUS, ENG, UKR |
| Pymorphy          | dictionary        | MIT                         | Python                       | -       | +   | Library     | RUS, ENG, DEU |
| RussianMorphology | dictionary        | Apache License <sup>6</sup> | Java                         | -       | +   | Library     | RUS           |
| RussianPOSTagger  | dictionary        | GPL <sup>7</sup>            | Java                         | +       | +   | GATE module | RUS           |

<sup>4</sup> The least standard public license GNU (GNU Lesser General Public License, LGPL)

<sup>5</sup> MIT License – free software license developed by Massachusetts Institute of Technology

<sup>6</sup> Apache license (English *Apache License* [Note 11](#)) — a license for free software of Apache Software Foundation.

<sup>7</sup> Standard public license GNU (GNU General Public License, GPL)



|             |                          |                    |                                |   |   |  |   |
|-------------|--------------------------|--------------------|--------------------------------|---|---|--|---|
| Snowball    | Porter algorithm         | BSD <sup>8</sup>   | GNU/Linux, Microsoft Windows   | + | + |  | RUS, ENG, DEU, FRA, ESP, POR, ITA, POU, SWE, NOR, DAN, FIN, HUN, TUR, ARM, BAG, CAT |
| Stemka      | dictionary               | Proprietary        | GNU/Linux, Microsoft Windows   | + | + |  | RUS, ENG  |
| SVMTTool    | supporting vector method | LGPL               | Perl                           | - | + | Library  | RUS, ENG  |
| TreeTagger  | decision trees           | Noncommercial      | GNU/Linux, Microsoft Windows   | + | + |  | RUS, ENG, DEU, FRA, ESP, POR, ITA, NLD, LAT, CHI, EST, SWA                          |
| FreeLing    | dictionary               | Conditionally paid | GNU/Linux                      | - | + | Library  | RUS, ENG, POR, ITA, ESP, CAT, GLG, CYM  |
| Proprietary |                          |                    |                                |   |   |  |   |
| RCO         | dictionary               | Commercial         | Microsoft Windows              | - | + | Package for DBMS Oracle RCO                      | RUS   |
| Solarix     | dictionary               | Commercial         | GNU/Linux, Microsoft Windows   | + | + | SDK module of the Russian grammatical dictionary | RUS, ENG  |
| Morpher     | dictionary               | Commercial         | Microsoft Windows, Web service | + | + | Web service/ Library                             | RUS, UKR  |
| ORFO        | dictionary               | Commercial         | Microsoft Windows              | - | + | Library  | RUS, ENG, DEU, FRA, ESP, POR, ITA, UKR  |

<sup>8</sup> BSD license ([English](http://en.cppreference.com/w/cpp/string/basic/basic_string_view) BSD license, Berkeley Software Distribution license) – the license agreement, first used for distribution of UNIX-like BSD operating systems.

Let's give a brief description of the most popular programs of morphological analysis.

### Snowball

Snowball is a mini-language to process lines and generate stemmers. It includes ready stemmers for English, French, Spanish, Portuguese, Italian, German, Danish, Swedish, Norwegian, Finnish and Russian languages. Stemming rules are described in Snowball language, translated into code in C or Java and then used as a conventional library. The development of Snowball is led by Martin Porter, a developer of one of the first such algorithms for the English language [14, 15].

Stemmer created by Snowball is a module source code in C or Java, implementing stemming rules for any language. There is a library for convenient use of the resulting module, which can include several stemmers when assembled.

The package includes source texts of libraries for C and Java, source texts of stemming rules for the above languages, the source code of demo program.

### Mystem

*Mystem* is a freeware morphological analyzer of the Russian language for non-commercial use from Yandex. The morphological analyzer works as a standalone application written in C. The program works with text files retrieving the information for morphologization or with standard I/O of words. Morphological analyzer shows all possible forms of the original words. The result of stemmer operation is a set of hypotheses for a non-existent word or one hypothesis for a vocabulary word. In addition, the latest version of Mystem for each variant of original form offers all the grammatical information (also synthesized for non-existent words) and frequency of its use in IPM (instances per million) (if the frequency is unknown – 0.00 is shown) – these data can be used in the future to select one normal form of a variety proposed by the program<sup>9</sup> [16-18].

### PHPMorphy

*PHPMorphy* is a freeware library for morphological analysis implemented on PHP platform. PHPMorphy allows performing the following tasks [19]:

- Lemmatization (getting normal form of

the word);

- Getting all word forms;
- Getting grammatical information for the word (part of speech, case, conjugation, etc.);
- Changing the word form according to predetermined grammatical characteristics;
- Changing the word form according to a given pattern.

Supported languages: English, Russian, German (AOT), Ukrainian, Estonian (based on Ispell). It is possible to add support for other languages using Myspell dictionary.

It supports various encodings: all single-byte (windows-1251, iso-8859-\*, etc.); Unicode encodings – utf-8, utf-16le/be, utf-32, ucs2, ucs4.

PHPMorphy uses the dictionary for operation. It supports freely distributed AOT and Myspell dictionaries, including for commercial use. Dictionaries are available in two forms: source and binary. The source dictionary is represented as XML file and contains word stems, change rules and grammatical information.

The article includes comparative analysis of Mystem and PHPMorphy morphological analyzers. The following experiment was conducted: the same text was entered at the input of PHPmorphy and Mystem morphological analyzers. The results of the experiment are given in Table 2. [20]

When using phpMorphy library, the case is possible when defining the part of speech, the function returns an array with multiple values for the word form.

For example – as shown in Figure 1.

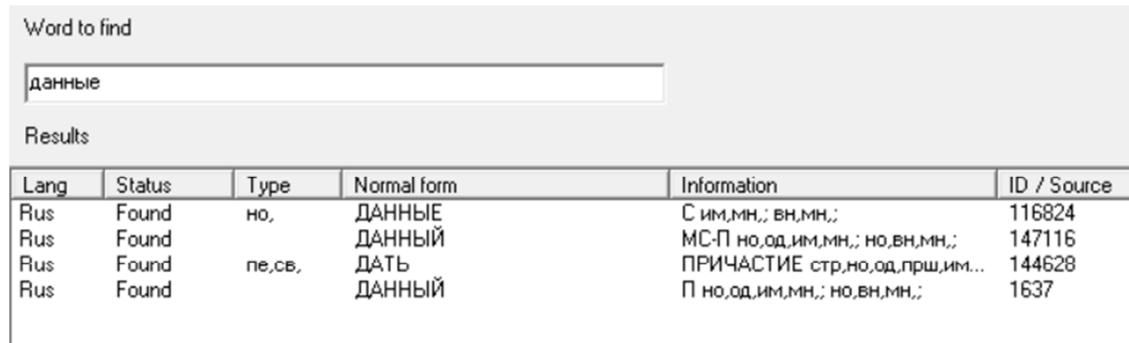
Therefore, one of the descriptors of the process of calculating the quantitative characteristics is the degree of unambiguous definition of parts of speech.

Comparative analysis of the experiment results showed that there are more ambiguities in automated definition of the part of speech when using Mystem morphologizator.

From Table 2 it is easy to see that in case of automated definition of parts of speech, there are more ambiguities when using Mystem morphologizator. Therefore, it is recommended to use PHPMorphy morphologizator as the automated module to determine parts of speech of

<sup>9</sup> However, in this case the algorithm stumbled on the word "Скоропечатник" (fast typographer) – a typewriter of Mikhail Ivanovich Alisov  
[[http://adm.rkursk.ru/index.php?id=13&mat\\_id=15963](http://adm.rkursk.ru/index.php?id=13&mat_id=15963)]

inflected languages (for bringing the word to the normal form).



Word to find

данные

Results

| Lang | Status | Type   | Normal form | Information                   | ID / Source |
|------|--------|--------|-------------|-------------------------------|-------------|
| Rus  | Found  | но,    | ДАННЫЕ      | С им,мн.; вн,мн.;             | 116824      |
| Rus  | Found  |        | ДАННЫЙ      | МС-П но,од,им,мн.; но,вн,мн.; | 147116      |
| Rus  | Found  | не,св. | ДАТЬ        | ПРИЧАСТИЕ стр,но,од,прш,им... | 144628      |
| Rus  | Found  |        | ДАННЫЙ      | П но,од,им,мн.; но,вн,мн.;    | 1637        |

Figure 1. Array with Multiple Values

Table 2 – Quantitative Characteristics of the Text

| Part of speech | phpMorphy   |   | MyStem                     |                          |
|----------------|---|---|----------------------------|--------------------------|
|                | Unambiguous interpretation (Issuing one semantic meaning) | Ambiguous interpretation (Array with multiple values) | Unambiguous interpretation | Ambiguous interpretation |
| Verbs          | 9   | 5   | 10                         | 6                        |
| Nouns          | 15  | 4   | 12                         | 12                       |
| Adjectives     | 10  | 3   | 8                          | 4                        |

#### 4. SOLUTION SELECTION

There are thousands of natural languages in the world, and analysis of the morphology of the existing languages would require considerable efforts. Therefore, the scope of the study was limited to the methods and algorithms for processing texts in Russian and Kazakh, as well as to the foreign developments that allow morphological analysis of texts in Russian, English and Turkic languages.

In describing the morphological analyzers in the previous section, we mainly studied the sources relating to their implementation for inflected languages (including Russian and English). Some of them are commercial systems, others are available for free use. In this paper, we mainly focused on the latter.

The morphological dictionary analyzers are universal for inflected languages and do not depend on specific languages. The progress of morphological analysis can be customized to the features of a certain language by setting various

rules in the source texts of the dictionary. It is also associated with the peculiarities of inflected languages where any word form is used with one or two affixes. Bringing these word forms to the dictionary form is straightforward. Each lemma in the dictionary gets the index of paradigm type which refers to the list of paradigm samples. Paradigms themselves are small in inflected languages, but their number is large. The analyzer builds a complete paradigm for each word, and compares the word form found in the text with this full paradigm.

If we are talking about the agglutinative languages, the situation changes slightly. Agglutinative word form is formed by joining unambiguous standard affixes to the stem in a strict order (i.e. one affix expresses one grammatical feature); morphemes have distinct borders, phonetic changes at the junction of morphemes are subject to strict rules. Word forms in agglutinative languages may contain a significant number of word forms of morphological features and thousands of words





may be formed from each stem. This is due to the large number of inflectional affixes.

The Kazakh language is an agglutinative language and, given the above, it is necessary to build a morphological analyzer that takes into account all the possible combinations of morphemes in the Kazakh language.

## 5. MORPHOLOGICAL MODEL OF THE KAZAKH LANGUAGE

The issues of automatic morphological analysis of word form and building a mathematical model of the text morphology are relevant to any natural language, including Kazakh as a representative of the group of Turkic languages. From the works of the Soviet period, we should highlight the articles of the linguist K.B. Bektaev [21], whose works formed the basis of applied linguistics of the Kazakh language. He was the first researcher to use mathematical methods to determine the information structure of the Kazakh language [22]. He has made the first Kazakh-Russian dictionary containing about 85 thousand words and phrases, Russian-Kazakh dictionary containing about 25 thousand words, more than 750 inflectional affixes indicating word synthesis algorithms [23]. Dictionaries of K.B. Bektaev have the necessary grammatical information (morphological, syntactic and semantic) to ensure the transfer of equivalent, variant and transformational translation equivalents.

Subsequently, the works of K.B. Bektaev were primarily used to create systems for machine translation from Kazakh.

Kazakhstan scientific school of applied linguistics is represented by two schools. First, the school of prof. A.A. Sharipbaev in the Eurasian National University ENU n.a. L.N. Gumilev [24]. Another school – in Almaty, headed by prof. A.A. Tukeev [25]. In the said paper of A. Sharipbaev [24], the processes of segmentation and generation of word forms are formalized, but there is no formal description of morphological and lexical sets of tags. The major scientific interest of the Kazakhstani researchers is represented by the field of segmentation and automation of the Kazakh language at the lexical and syntactic level. Another work of A.A. Sharipbaev [26] is of great interest as an attempt of hardware implementation of generation of the words in Kazakh using associative memory device.

To understand the possibilities of application of the above methods of normalization to various

languages, it is necessary to consider the linguistic classification of the Kazakh language. In terms of types of morphological structure, the Kazakh language is agglutinative (morphemes are semantically separated but really joined in words). Agglutinative languages (in this case, the Kazakh language) are characterized by a well-developed system of word-formative and inflection affixation, unambiguous grammatical affixes, and the lack of alternations.

In the Kazakh language being agglutinative, new words and different forms of words are formed by consecutive joining of word-formative and morphogenetic affixes and inflections to the root or stem of the word. Each affix has only one grammatical meaning, and each grammatical meaning is always expressed by the same affix. Suffixes and inflections are dependent on the softness and hardness of vowels, eg.: кiтaп – кiтaптaр, дaптeр – дaптeрлeр. The root in the Kazakh language remains unchanged, affixes harmonize with the root, that is, words are formed according to the law of vowel harmony – the law of combination of sounds of the main part of the word and affixes [7, 8, 23].

New word forms are formed taking into account morphological and semantic features of the original forms as follows: first, suffixes are added to the original word form. Then, moving from left to right, the category (flat, ringing, etc.) of the last letter (the last sound) of the original word form is determined to add a certain ending [8].

General morphological form of determining the composition is as follows [7, 8]:

Түбір (root) + жұрнақ (suffix) + жалғау (ending) (1).

Based on the analysis and grammar of the Kazakh language, the following basic rules of the Kazakh language can be distinguished [7,8]:

– In the Kazakh language, the word can not end with a voiced consonants "б", "в", "г", "ф", "д", "ж". There are exceptions in this language where the suffix beginning with a vowel is removed, and letters "б", "г", "ф" in the end are converted into the following letters: "п", "к", "к". For example, the letter "п" into "б", "к" into "ф", "к" into "г".

– A hard syllable is followed by a hard ending, a soft syllable is followed by a soft ending.



– The softness and hardness of words in the Kazakh language is determined by the presence of a certain vowel in the last syllable of the word. For example, the word is hard, if there are vowels а, о, ұ, ы, я; and it becomes soft, if there are vowels ә, ө, ү, і, е. The hardness or softness of the words also correlates with the presence of some consonants: a word is hard if it contains consonants к and ғ, and it is soft if there are к and г.

– Each subsequent ending depends on the previous one by several parameters. By hardness: if the last syllable of the word is hard, each subsequent ending will be hard, because the hardness of each subsequent ending depends on the previous one. Thus, if a word is hard, all endings are hard, and if it is soft, all endings are

soft.

As is known, the morphemes are the smallest meaningful (semantic) units of language, composing a word form, and then the lexical unit respectively. In the Kazakh language, endings are divided into four types. The below endings will be directly used in the developed algorithm to determine a word stem.

- Denote the following sets of endings (affixes) as  $P_i$ , for  $i=1, 2, 3, 4$ .
- $P_1$  – a set of three-letter endings (ending of a plural);
- $P_2$  – a set of endings (possessive endings);
- $P_3$  – a set of endings (personal endings);
- $P_4$  – a set of endings (case endings).

Table 3 below describes determinations of morphemic composition ( $P_i$ , where  $i=1, 2, 3, 4$ ).

Table 3 – Endings in the Kazakh language

| No. | Types of endings           | Endings  |
|-----|----------------------------|--|
| 1.  | Ending of a plural - $P_1$ | 'лар', 'лер', 'дар', 'дер', 'тар', 'тер'   |
| 2.  | Possessive endings - $P_2$ | 'ым', 'ім', 'м', 'ың', 'ің', 'н', 'ыңыз', 'іңіз', 'ңыз', 'ңіз', 'сы', 'сі', 'ы', 'і', 'ымыз', 'іміз', 'мыз', 'міз'   |
| 3.  | Personal ending - $P_3$    | 'мын', 'мін', 'бын', 'бін', 'пын', 'пін', 'сың', 'сің', 'сыз', 'сіз', 'мыз', 'міз', 'быз', 'біз', 'пыз', 'піз', 'сыңдар', 'сіңдер', 'сыздар', 'сіздер', 'м', 'н', 'ңыз', 'ңіз', 'к', 'к', 'ндар', 'ндер', 'ңыздар', 'ңіздер' |
| 4.  | Case endings - $P_4$       | 'ның', 'нің', 'дың', 'дің', 'тың', 'тің', 'ға', 'ге', 'қа', 'ке', 'ны', 'ні', 'ды', 'ді', 'ты', 'ті', 'да', 'де', 'та', 'те', 'нан', 'нен', 'тан', 'тен', 'дан', 'ден', 'мен', 'бен', 'пен'                                  |

For convenience of implementation, systematization of endings was studied, and the order of the rules is as follows:

- A – Ending of a plural + Case ending
- B – Plural + Personal ending.
- C – Plural + Possessive ending.
- D – Plural + Possessive ending + Case ending.
- E – Plural + Possessive ending + Personal ending.
- F – Plural + Case ending+ Personal ending.
- G – Plural + Possessive ending + Case ending+ Personal ending.

Let:  
 H is a set of word forms;  
 I is a set of normal word forms;  
 Represent each word  $z$  as  $z = y \wedge x$  as a concatenation of two (or more) words  $y$  and  $x$ ;  
 If word  $x \in P_i$ , then denote as  $P_i(x)$  for all  $i = 1, \dots, 4$ ;

If word  $x \in I$ , then denote as  $I(x)$ ;  
 If word  $x \in H$ , then denote as  $H(x)$ .

Then rules A – G of analytical separation of the stem by steps satisfy the following formulas:

Let an arbitrary word  $z = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_k$ , where  $x_i$  is the maximum number of letters in in the ending of word  $z$ . Let  $i = k, x = x_i$ .

Step 1

$$A = \begin{cases} \text{if } P_4(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \text{then let } z = z \setminus x, i = i - 1. \\ \text{if } P_1(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \text{then let } z = z \setminus x, i = i - 1 \end{cases}$$

Step 1 is checked for applicability (conditions of compatibility), and if it is not applicable, then go to step 2.



Step 2.  $G$

$B$

$$= \left\{ \begin{array}{l} \text{if } P_3(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1. \\ \text{if } P_1(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \end{array} \right. = \left\{ \begin{array}{l} \text{if } P_3(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1. \\ \text{if } P_4(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \\ \text{if } P_2(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \\ \text{if } P_1(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \end{array} \right.$$

Step 3.  $C$

$$= \left\{ \begin{array}{l} \text{if } P_2(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1. \\ \text{if } P_1(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \end{array} \right.$$

Step 4.  $D$

$$= \left\{ \begin{array}{l} \text{if } P_4(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1. \\ \text{if } P_2(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \\ \text{if } P_1(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \end{array} \right.$$

In carrying out steps 1-7, before every action there is a search of the current word in the list of exceptions; if the word is an exception, the actions with it are ceased. For example, the word "қағар" (threat) and "сымсыз" (wireless) should be classified as exceptions as "реп" and "сыз" in them are not endings.

Step 8.

If at the end of the word there is a letter  $\Gamma$ , it shall be replaced by  $\kappa$ ; similarly  $\Gamma - \kappa$  and  $\delta - \pi$ . This requirement is imposed by a law of vowel harmony in the Kazakh language.

Step 5.  $E$

$$= \left\{ \begin{array}{l} \text{if } P_3(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1. \\ \text{if } P_2(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \\ \text{if } P_1(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \end{array} \right.$$

Step 9.

if  $\bigwedge_{i=1}^4 \neg P_i(x)$ , then complete the task

At the output we get the stem of the analyzed word form.

The following table 4 shows the use of the given algorithm on a few examples of the Kazakh language. Word forms in Russian and English are normalized using PHPMorphy library.

Step 6.  $F$

$$= \left\{ \begin{array}{l} \text{if } P_3(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1. \\ \text{if } P_4(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \\ \text{if } P_1(x) \wedge H(z \setminus x), \text{ where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge \dots \wedge x_{i-1} \\ \quad \text{then let } z = z \setminus x, i = i - 1 \end{array} \right.$$

When normalizing the word form "Рұқсат етілген ақпараттық ағын" it is brought to the form "Рұқсат ету ақпараттық ағын", and in this case suffix "ілген" is removed from "етілген" term and suffix "+y" is added to the word stem, in accordance with the rules of normalization of verbs in the Kazakh language.

Comparing the mechanism for normalization in Russian and Kazakh languages, we would like to note that normalization process in Kazakh is a bit easier than in Russian, because after cutting off the endings, no other endings are added to the stem (with the exception of vowel harmony). This is

Step 7.



clearly seen in the second example in Table 4, where the phrase "Деректерді өңдеу бағдарламаларының орындалу кестесі" is converted to a normal form "Дерек өңдеу бағдарлама орындалу кесте". For example, the noun "бағдарламаларының" by cutting off the case ending "-ның", possessive ending "ы", and then ending of a plural "-лар", is reduced to the normal form "бағдарлама".

In the third example, when the phrase "Ақпараттық жүйелердің сенімділігі" is brought to the normal form "Ақпараттық жүйе сенімділік", it is shown how to use the law of vowel harmony in the Kazakh language: in the term "сенімділігі", after truncation of the possessive ending "-і", according to the above algorithm the check at step 8 is conducted. In this case, by replace letter "г" to "к", we have the following normal form "сенімділік".



Table 4 – Word Form Normalization

| Word form                                       | Lemma<br>(normal word form)                      | Word form                                      | Lemma<br>(normal word form)                  | Word form   | Lemma<br>(normal word form)           |
|---|--|--|--|---|---------------------------------------|
| Russian (PHPMorphy)                             |  | English (PHPMorphy)                            |  | Kazakh (our algorithm)                              |                                       |
| Автоматическое извлечение метаданных            | Автоматический извлечение метаданные             | Automatic metadata extraction                  | Automatic metadata extraction                | Метадеректерді автоматты алу                        | Метадерек автомат алу                 |
| Интеграция данных                               | Интеграция данные                                | Data integration                               | Data integration                             | Деректер интеграциясы                               | Дерек интеграция                      |
| Структуры данных                                | Структура данные                                 | Data structure                                 | Data structure                               | Деректер құрылымы                                   | Дерек құрылым                         |
| Расписания выполнения программ обработки данных | Расписание выполнение программа обработка данные | Schedules performance data processing programs | Schedule performance data processing program | Деректерді өңдеу бағдарламаларының орындалу кестесі | Дерек өңдеу бағдарлама орындалу кесте |
| Надежность информационных систем                | Надежность информационный система                | Reliability of information systems             | Reliability information system               | Ақпараттық жүйелердің сенімділігі                   | Ақпараттық жүйе сенімділік            |
| Политики информационной безопасности            | Политика информационный безопасность             | Information security policies                  | Information security policy                  | Ақпараттық қауіпсіздік саясаттары                   | Ақпараттық қауіпсіздік саясат         |
| Учет информационных ресурсов                    | Учет информационный ресурс                       | Accounting of information resources            | Accounting information resource              | Ақпараттық ресурстарды есепке алу                   | Ақпараттық ресурс есеп алу            |
| Алгоритм автоматического выделения основ        | Алгоритм автоматический выделение основа         | Algorithms of allocation basics                | Algorithm allocation basics                  | Түбірлерді автоматты белгілеудің алгоритмі          | Түбір автоматты белгілеу алгоритм     |
| Нормализация слов                               | Нормализация слово                               | Normalization of words                         | Normalization word                           | Сөздердің нормалануы                                | Сөз нормалану                         |
| Классификация информационных ресурсов           | Классификация информационный ресурс              | Classification of information resources        | Classification information resource          | Ақпараттық қорлар классификациясы                   | Ақпараттық қор классификация          |
| Доступ к информационным ресурсам                | Доступ информационный ресурс                     | Access to information resources                | Access information resource                  | Рұқсат етілген ақпараттық ағын                      | Рұқсат ету ақпараттық ағын            |

**6. PRACTICAL IMPLEMENTATION AND TESTING OF THE ALGORITHM**

As a platform for implementation of the morphological analyzer, we used “Electronic library management system” developed in ICT SB RAS [27, 28].

Adding termNormName (Field) element in thesaurus metatable (Zthes data scheme) will automatically allow to bring the word form expressing the term to the normal form (singular,

nominative case), by connecting the module of morphological analyzer working with the word forms in three languages (Russian, Kazakh and English).

For example, in the Kazakh language, for the word form "Ақпарат құпиялылығының иерархиялық емес категориясы" the algorithm generates the normal form "Ақпарат құпиялылық иерархиялық емес категория" (see Figure 2).

db4.sbras.ru/elbib/data/admin/show.phtml#menu

**Просмотр документа N 2173 в коллекции zthes\_cat (13)**

| NN | Имя элемента   | Название элемента             | Значение элемента                                      |
|----|----------------|-------------------------------|--|
| 1  | title          | Название термина              | Ақпарат құпиялылығының иерархиялық емес категориясы    |
| 2  | link_id        | UID Основной link_id (termID) | 8D9120F8   |
| 3  | term_qualifier | term_qualifier                | Ақпарат құпиялылығының иерархиялық емес категориясы_KZ |
| 4  | field          | Нормальная форма термина      | Ақпарат құпиялылық иерархиялық емес категория          |

Figure 2. Normal Form for the Term in Kazakh

For the word form in Russian: "Расписания выполнения программ обработки данных" the algorithm generates the normal form "Расписание

выполнение программа обработка данные" (see Figure 3).

db4.sbras.ru/elbib/data/admin/show.phtml#menu

**Просмотр документа N 1565 в коллекции zthes\_cat (13)**

| NN | Имя элемента   | Название элемента             | Значение элемента                                  |
|----|----------------|-------------------------------|--|
| 1  | title          | Название термина              | Расписания выполнения программ обработки данных    |
| 2  | link_id        | UID Основной link_id (termID) | DAC1BE20   |
| 3  | term_qualifier | term_qualifier                | Расписания выполнения программ обработки данных_RU |
| 4  | field          | Нормальная форма термина      | Расписание выполнение программа обработка данные   |

Figure 3. Normal Form for the Term in Russian

For the word form in English "Schedules performance data processing programs" the algorithm generates the normal form "Schedule

performance data processing program" (see Figure 4).

db4.sbras.ru/elbib/data/admin/show.phtml#menu

### Просмотр документа N 1565 в коллекции *zthes\_cat* (13)

| NN | Имя элемента   | Название элемента             | Значение элемента                                 |
|----|----------------|-------------------------------|---|
| 1  | title          | Название термина              | Schedules performance data processing programs    |
| 2  | link_id        | UID Основной link_id (termID) | 06D2C1BF  |
| 3  | term_qualifier | term_qualifier                | Schedules_performance_data_processing_programs_EN |
| 4  | field          | Нормальная форма термина      | Schedule performance data processing program      |

Figure 4. Normal Form for the Term in English

In computer verification of the algorithm, 1,500 arbitrary word forms in IT technology in the Kazakh language were chosen.

It resulted in 100% of correctly generated normal forms, that is, we can say that the algorithm works correctly.

## 7. CONCLUSION

A new algorithm was proposed to consider, for normalization of word forms in the Kazakh language, which showed quite good results when tested on word forms in IT technology. On the basis of the above algorithm, a morphological analyzer was developed integrated in PHPMorphy library and connected to the integrated distributed system SUEB in NSU. Note that all thesauri in SUEB support Zthes data scheme, and 3 new rules were added for thesaurus on IT technology in the Kazakh language, allowing implementing the normalization algorithm proposed by the authors.

## REFERENCES:

- [1] D. Knuth, "The Art of Computer Programming", *Sorting and Searching, Second Edition*, Massachusetts: Addison-Wesley, 1998. ISBN 0-201-89685-0.
- [2] A.I. Mikhailov, A.I. Chernyi and R.S. Gilyarevskii, "Fundamentals of Informatics", Moscow: Nauka, 1968.
- [3] A.I. Mikhailov, A.I. Chernyi and R.S. Gilyarevskii, "Scientific Communications and Informatics", Moscow: Nauka, 1976.
- [4] Yu.I. Shokin, A.M. Fedotov and V.B. Barakhnin, "Problems of Information Retrieval", Novosibirsk: Nauka, 2010.
- [5] Yu.M. Arskii, R.S. Gilyarevskii, I.S. Turov, and A.I. Chernyi, "Infosphere: Information Structures, Systems, and Processes in Science and Society", Moscow: VINITI, 1996.
- [6] A.I. Rakitov, "Encyclopedia of Philosophy", vol. 5, Moscow: Sovetskaya Entsiklopediya, 1970, p. 298.
- [7] Kazakh grammar. Phonetics, word formation, morphology, syntax, Astana, 2002.
- [8] M. Balakaev, Modern Kazakh, Astana, 2006.
- [9] A.M. Fedotov, O.L. Zhizhimov, O.A. Fedotova, V.B. Barakhnin. "A model of information system to support scientific and educational activities", *Vestnik NSU Series: Information Technologies*, vol. 12, no 1, 2014, pp. 89-101. ISSN 1818-7900.
- [10] S.R. Ranganatan, *Colon Classification*, 6th ed., Bombay: Asia, 1963.
- [11] V.B. Barakhnin, A.M. Fedotov, "Building models of documentary and factographic retrieval in digital libraries", *Automatic Documentation and Mathematical Linguistics*. vol.48, no. 6, 2014, pp. 296-304. ISSN 0005-1055, EISSN 1934-8371.
- [12] Yu.A. Shreider, Equality, Similarity, Order, Moscow: Nauka, 1971.
- [13] G. Salton, Dynamic information and library processing. N.J.: Prentice Hall, 1975.
- [14] M.F. Porter, "An algorithm for suffix stripping", *Program*, vol. 14, no. 3, 1980, pp. 130-137.
- [15] P. Willett, "The Porter stemming algorithm: then and now", *Program: Electronic Library and Information Systems*, vol. 40, no. 3, 2006, pp. 219-223.
- [16] Segalovich, "A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine", 2003, pp. 273-280.



- [17] I.V. Segalovich, M.A. Maslov, "Russian morphological analysis and synthesis with generation of models of inflection for words not described in the dictionary", Moscow: Dialog, vol. 2, 1998, pp. 547-552.
- [18] Mystem morphological analyzer of text in Russian [e-resource]; Yandex Company [site], 2003-2013, URL: <http://company.yandex.ru/technologies/mystem/>
- [19] Library phpMorphy, URL: <http://phpmorphy.sourceforge.net>
- [20] A.A. Rybanov, "Automated determination of quantitative characteristics of text", *Modern scientific research and innovations*, vol. 34, no. 2, 2014, p. 5.
- [21] K.B. Bektaev, *Statistical and information typology of Turkic text*. Almaty, 1978, p.183.
- [22] K.B. Bektaev, R.G. Piotrovsky, "Mathematical methods in linguistics", Probability theory and simulation of language standard. Almaty: Publishing house of KazSU n.a. Kirov, 1973, 281 p.
- [23] K. Bektayev, Big Kazakh-Russian and Russian-Kazakh dictionary, Almaty: "Altyn Kazyna", 1999.
- [24] A.A. Sharipbayev, G.T. Bemanova, "Building logical semantics of the words in the Kazakh language", *Knowledge-Ontologies-Theories: Proc. of All-Russian Conf. with int. participation, October 3-5, 2011*, Novosibirsk, 2011.
- [25] U. Tukeev, D.R. Rakhimova, "Augmented attribute grammar in meaning of natural languages sentences", *SCIS-ISIS, The 6th International Conference of Soft Computing and Intelligent Systems, The 13th International Symposium on Advanced Intelligent Systems (November 20-24)*, Kobe, Japan, 2012, pp. 1080-1084.
- [26] A.A. Sharipbayev, G.T. Bekmanova, B.Zh. Ergesh, A.K. Buribaeva, M.Kh. Karabalaeva, "Intelligent morphological analyzer based on semantic networks", *Proceedings of the international scientific conference "Open Semantic Technology for Intelligent Systems" (OSTIS-2012)*, Minsk: BSUIR, February 16-18, 2012, pp. 397-400.
- [27] Y.I. Shokin, A.M. Fedotov, O.L. Zhizhimov, O.A. Fedotova, "The evolution of information systems: from websites to information resource management systems", *Vestnik NSU Series: Information Technologies*, vol., 13, no. 1, 2015, pp. 117-134. ISSN 1818-7900.
- [28] Y.I. Shokin, A.M. Fedotov, O.L. Zhizhimov, O.A. Fedotova, "Electronic library management system at integrated Distributed Information System of SB RAS", *Infrastructure of scientific information resources and systems: Collection of scientific articles of the Fourth All-Russian Symposium*, Moscow: Computing Center of the Russian Academy of Sciences, vol. 1, 2014, pp.11-39. ISBN: 978-5-19601-103-6.