EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 3, Issue 3 (2015) 69– 79

DEVELOPMENT AND IMPLEMENTATION OF A MORPHOLOGICAL MODEL OF KAZAKH LANGUAGE

A. Fedotov, J. Tussupov, M. Sambetbayeva, I. Idrisova, A. Yerimbetova

Abstract In this paper we considered morphological models and existing algorithms of the word normalization of natural languages. The study described algorithms for the automatic word basis for some natural languages and possible synthesis of a normal form of the word for the Kazakh language. The rules are developed for the normalization of the words for the Kazakh language and the algorithm for the processing of both dictionary words and words, which are not in the dictionary, including non-existing words. The thesaurus of scientific and technical terms on information technologies in the Kazakh language is created and the system of normalization is realized for it, proving the operability of the developed algorithm.

Key words: the morphology of the Kazakh language, normalization of words, algorithms of allocation basics, synthesis of a normal form.

AMS Mathematics Subject Classification: 03B65, 68T50, 91F20

Introduction

The Kazakh language is one of the Kipchak group of Turkic languages. It is a type of synthetic agglutinative languages[1]. It has a rich and complex morphology. Word is usually composed of a base and at least two or three affixes (suffix + ending).

The necessity of the word normalization (the creation of the morphological parser) appeared when operated at information-retrieval thesaurus with account of morphology of Kazakh language in the full-text base of information technology. In analyzed text the word normalization simplifies the work with it: such as indexation, the subsequent information research in the constructed index, and also the solution of tasks of classification (clustering) and automatic abstracting of documents of scientific and technical subject [1].

This article is devoted to the analysis of a problem of synthesis (search) of a word basis of the Kazakh scientific and technical terms and creation of algorithms of its decision. We will mark that orientation to the thesaurus strongly simplifies the considered problem.

Different languages have different semantic and grammatical features therefore the algorithms, which are successfully used for processing of one language, often show very low productivity in another language. For the analysis we took the morphological model which analyzes the greatest number of word forms from a normal form of a noun,

[1]Agglutinative language (from armor. agglutinatio – attaching) – the language, having a system, in case of which the dominating type of change is agglutination ("attaching") of different suffixes or prefixes and each of them has only one value

an adjective and a normal form of a verb as a component of scientific and technical terms of the Kazakh language.

However the complexities of processing of natural language sometimes allow to allocation problems which can already be solved by algorithmically way, for example: defining of the words of texts such as noun, adjective and verb forms or split the text into logical groups. However, some features of the natural languages significantly reduce the effectiveness of these solutions: for example, account of all word forms for each word in the Kazakh language in the order of increasing complexity of processing texts.

1 Review of existing approaches of morphological analysis

We will consider the methods and tools of morphological analysis in the problems of normalization of words in scientific and technical terms for the Kazakh language. There are two main different approaches to the morphological analysis: methods, based on dictionaries and non-dictionary morphological parser.

We will consider the morphological models, the approaches of creation the algorithms of word normalization. These approaches are divided into two classes: Stemming algorithms and lemmatization.

Stemming is the process of searching the word basis for the given initial word. The word basis not always coincides with the root of the word.

Lemmatization is the process of reducing a word (word form) to lemma (normal word form).

We will give some explaining definitions:

Lemma – the normal (dictionary, canonical) form of the word (for example, in the Kazakh language a lemma for scientific and technical terminology is:

1. Nouns – the Nominative case, singular,

2. Adjectives appear as determination, and don't acquire the terminations, and change of the adjectives appearing as nouns doesn't differ from change of nouns),

3. Verbs – the initial form of a verb.

The word form is the word provided in a certain grammatical form.

Lexeme is the word as the abstract unit of a natural language. In one lexeme are integrated different paradigmatic forms (word forms) of one word. For example ақпараттандыру (to report) – a lemma, ақпараттандырылған (information), ақпараттандырылатын (informed) – a lexeme.

The morphological analysis gives the solution of two main objectives:

– Tasks of analysis – determination of a normal form of the word on arbitrary word form,

– Tasks of synthesis – creation of all word forms on a normal form.

The majority of popular algorithms realize a lemmatization (word normalization) with the using of the word basis (algorithm of a stemming). However, there are two problems characteristic for the Kazakh language: in the first place, the synthesis of a normal form strongly depends on a method of receiving a basis of the word, and in the second place, the majority of implementations synthesizes all possible lemmas, without selecting from them the single result, or stops on determination of a basis of the word.

We will analyze two most popular algorithms of a lemmatization, which are based on different principles – The Porter stemming algorithm, algorithm of Yandex.

1.1 The Porter stemming algorithm

The Porter stemming algorithm was published in 1980 by Martin Porter for the English language. It was described by a sequence of steps, each of which, under certain rules may be one of certain reforms of ending.

This rule has the following structure [2]:

<Condition> <ending> <the new ending>.

The main idea of The Porter stemming algorithm is that there is a limited number of both form-building and word-formative suffixes, and a basis of the word will be transformed without use of any bases (dictionaries) of bases: only a set of the existing suffixes (thus difficult composite suffixes break on simple) and manually the given rules.

The fact, that Porter's algorithm doesn't use any dictionaries and bases, is a plus for high-speed performance and a range of application (he not bad copes with nonexistent words), and at the same time minus from the point of view of the accuracy of separation of a basis.

Besides, The Porter stemming algorithm has a minus such as a human factor: rules for check-up are set manually. The rules sometimes connected with grammatical features of language, increasing probability of an error.

For example we will give one of rules offered by Porter and used in the Kazakh word forms ($m > 0$) ЫЛҒАН $\mapsto$ У which means that if the word form has this kind of an ending, the ending is replaced on -"у". For example, application of this rule to a word form "ақпараттандырылған" leads it to a normal form to «ақпараттандыру».

1.2 Algorithm of Yandex

The algorithm of Yandex (Mystem) is a development of Ilya Segalovich (Yandex, 1998) [4]. This algorithm of the morphological analysis is dictionary. The main feature of this algorithm is that for the word form, which isn't described in the dictionary or the nonexistent word, the algorithm generates its presumable model of the word change – one or several variants of a normal form of the word. (For example, the stakelkakh stakelkh? |stakelka? |stakelok?), then, replenishing the dictionary with new lexemes, the generated hypothetical articles can be saved in this dictionary (or in other dictionary of the same type) for further using [5].

The algorithm processes an analyzable text word-by-word. Each word is checked for accessory to the list of lexically dependent words, which don't have nominative functions in language , an example in the Kazakh language: және (and), әрі(any), да(and), ғой, қана(only) etc. This list includes pretexts, particles, some interjections and most frequently used adverbs of the Kazakh language. Besides, all words having length less than three characters are also not processed. If the word is lexically the dependent word, it registers in a line of result without changes, the algorithm passes to processing of the following word.

The following action is searching for the word which is an independent part of speech. The algorithm of separation of a word basis is launched for each word. By means of a suffixes tree the suffix is split off from the word, and then there is a search of an estimated basis in a bases tree. If the basis is in a tree – we check, whether the combination of this basis and this suffix is possible, whether the received model of a necessary part of speech satisfies. If yes – the lemma, corresponding to this model, returns.

Pluses of this algorithm for each option of a normal form offers all grammatical information (synthesizable and for nonexistent words), these data can be used further for a choice of one normal form from the set offered by the program.

Minuses of this algorithm are that in the absence of the entered word it can not always cope with this task. Also he doesn't cope with the diminutive-hypocoristic form of the word. As an example we approved the word "машинка" – the algorithm didn't cope with this task, thereby it was necessary to add this phrase to the user dictionary (– fixlist). In other words there is a suspicion that the algorithm not always copes with words which aren't present in the dictionary. For example, the word "Скоропечатник" was checked.

2 Desicision making

As it was already stated above, there are two essentially differing approaches to the morphological analysis. The first method, which we considered, is Porter's algorithm, which works in the nondictionary mode. The second method is the algorithm from Yandex, working with the dictionary. The problem of the morphological analysis is solved more complete by the dictionary analyzers. It allows defining grammatical characteristics for word usage in texts in natural languages. It becomes impossible to carry out, for example, fragmentation of texts, separation of noun and prepositional phrases, homogeneous parts of the sentence without these characteristics. However this method by using the dictionary has one principal shortcoming – if the analyzable word form doesn't present in the dictionary, it is impossible to receive any morphological information on it.

Therefore for a solution of the problem of the morphological analysis at creation of a search index for full text search the task to develop the analytical methods of allocation of bases using linguistic dictionaries is set. Methods of this kind don't completely solve a problem of the morphological analysis: it is difficult to define a part of speech and grammatical features of word forms. However analytical algorithms are effective for problems of indexation of text massifs, creation of procedures of work with dictionaries of scientific and technical terms for the Kazakh language.

For the understanding of opportunities of application of the considered analysis methods to different languages it is necessary to consider linguistic classification of the Kazakh language. From the point of view of types of morphological structure the

Kazakh language is agglutinative language (morphemes are semantic separated, but are really united in words). For agglutinative languages (in this case, for the Kazakh language) rather developed system of word-formation and inflectional affixation, grammatical uniqueness of affixes, absence of alternations is characteristic.

Proceeding from absence of this kind of algorithm essentially to the Kazakh language were developed model and rules of algorithm for the Kazakh language.

3 Morphological model of Kazakh language

In the Kazakh language, word forms are formed by concatenating of the roots and affixes (suffixes and endings). Thereby, each affix is associated with a set of semantic features and the order of addition of the affixes is strictly defined. For example, for nouns suffix firstly adds to the base of the word and then the plural form ending, then the possessive ending, followed by the case ending, and only after that – the ending of the conjugation form (adds only to animated nouns) [6].

New word forms are formed taking into account the morphological and semantic features of the initial forms by the following way: first, suffix adds to the initial form of the word. Then, by moving from left to right, there would determine the category of (unvoiced, sonant, etc.) the last letter (the last sound) of the initial form of a word to add this or that end (ending). [7]

General morphological form of determination of the composition looks like this: Түбір (root) + жұрнақ (suffix) + жалғау (ending)[8].

By virtue of the analysis and grammar of the Kazakh language we can distinguish the following basic rules of the Kazakh language. [9]:

- In the Kazakh language, word can't be end in voiced consonants as: "б", "в", "г", "ғ", "д", "ж". In this language, there are taking place exceptions in which suffix removes which begins with a vowel, and "б", "г", "ғ" standing at the end transforms into the following letters: "п", "қ", "к". For example, the letter "п" to "б", letter "қ" to "ғ", letter "к" to "г".

- After a solid syllable should be a solid end, after a soft syllable should be a soft end.

- The softness and hardness of words in Kazakh language are determined by the presence of a certain vowel in the last syllable of the word. For example, the word is a solid, 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 correlates well with the presence of some consonants: a word considers as hard if it contains consonants ғ and қ; and soft, if it contains к and г.

- Every following ending depends on the previous one by several parameters. On solidness: if the last syllable of the word were solid, then each following ending would be solid, because the solidness of the following ending depends on the previous one. Thus, if word is solid, then all endings are solid; if it soft, then endings are soft.

It is known that morphemes are the smallest meaningful (semantic) units of language, which are composes word forms, and then, respectively, the lexical items. In the Kazakh language endings are divided into four types. Endings which are described at the bottom, straightly, will be used in developing algorithm of determining of basis of the word.

Let's denote by P_i – the following sets of endings (affixes), for $i = 1, 2, 3, 4$.

P_1 – a plurality of three-letter endings (the plural ending);

P_2 – a plurality of endings (possessive ending);

P_3 – a plurality of endings (personal ending);

P_4 – a plurality of endings (case ending).

Below in the tables 1, determination morphemic composition is described (P_i, where $i = 1, 2, 3, 4$):

Table 1.

#	Ending Types	Endings
1	Plural endings – P_1	'лар', 'лер', 'дар', 'дер', 'тар', 'тер'
2	Possesive endings – P_2	'ым', 'ім', 'м', 'ың', 'ің', 'ң', 'ыңыз', 'іңіз', 'ңыз', 'ңіз', 'сы', 'сі', 'ы', 'і', 'ымыз', 'іміз', 'мыз', 'міз'
3	Personal endings – P_3	'мын', 'мін', 'бын', 'бін', 'пын', 'пін', 'сың', 'сің', 'сыз', 'сіз', 'мыз', 'міз', 'быз', 'біз', 'пыз', 'піз', 'сыңдар', 'сіңдер', 'сыздар', 'сіздер', 'м', 'ң', 'ңыз', 'ңіз', 'қ', 'к', 'ңдар', 'ңдер', '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'

Taking into account all ending compounds combination of an affixal groups of the Kazakh language, have been identified more than 750 inflectional affixes with an indication of synthesis algorithms of words that provides creative possibilities to increase the number of words and phrases to use [8].

For the convenience of implementation, there is ending systematization was investigated and order of the rules has the following form:

A – Plural ending + Case ending

B – Plural form + Personal ending.

C – Plural form + Possessive ending.

D – Plural form + Possessive ending + Case ending.

E – Plural form + Possessive ending + Private ending.

F – Private ending + Plural form's ending.

G – Possessive ending + Case ending.

H – Possessive ending + Personal ending.

Let us assume that:

Q – an arbitrary a monadic predicate;

W – a plurality of the normal form the word;

Each word z represented in the form $z = y \wedge x$ as the concatenation of two (or more) words y and x;

If word $x \in P_i$, then let donate it for all as $i = 1, ..., 4$;

If word $x \in W$, then let donate it as;
If word $x \in Q$, then let denote $Q(x)$.

Then our rules A–H of the analytical highlight of the base to the steps satisfy the following formulas:

Let us assume that an arbitrary word, where x_i the maximum number of letters in the word's end. We put $i = k, x = x_i$.

Step 1.

$$A = \begin{cases} P_4 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_1 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \end{cases}$$

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

Step 2.

$$B = \begin{cases} P_2 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_1 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \end{cases}$$

Step 3.

$$C = \begin{cases} P_4 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_3 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_1 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \end{cases}$$

Step 4.

$$D = \begin{cases} P_2 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_3 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_1 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1}) \\ \text{We put } z = z \setminus x,\ i = i-1 \end{cases}$$

Step 5.

$$E = \begin{cases} P_1 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_2 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \end{cases}$$

Step 6.

$$F = \begin{cases} P_4 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_3 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \end{cases}$$

Step 7.

$$G = \begin{cases} P_2 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_3 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \end{cases}$$

Step 8.

$$H = \begin{cases} P_3 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \\ P_1 \mapsto Q(z \setminus x)), \text{where } z \setminus x = x_0 \wedge x_1 \wedge x_2 \wedge ... \wedge x_{i-1} \\ \text{We put } z = z \setminus x,\ i = i-1 \end{cases}$$

Application of the selected rule (Step 1-8), and check its final symbol, depending on which algorithm is either stops or proceeds to the step 1.

Step 9.

$$\bigwedge_{i=1}^{4} \neg P_i(x) \mapsto W(z)$$

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

Here is an example of written algorithm test. At the bottom side of Table 2 described sample of the dictionary where words without last k letters for word forms, which are made by using of our algorithm.

Table 2. The dictionary of scientific and technical terms

A token for the Kazakh language (Examples of the single word forms.)	A token for the Russian language (Examples of the single word forms.)	The basis of the word	Lemma (The normal form of the word)
Ақпарат	информация	ақпарат	информация
ақпаратпен	информацией	ақпарат	информация
Ақпаратқа	информации	ақпарат	информация
Ақпаратты	информации	ақпарат	информация
ақпараттық	информационный	ақпаратт	информационный
ақпараттандыру	информатизация	ақпараттандыру	информатизация
ақпараттандырылған	информационный	ақпараттандыру	информационный
Жүйе	система	жүйе	система
Жүйенің	системы	жүйе	система
Жүйелік	системный	жүйелік	системный
жүйелерді	систем	жүйе	система
Жүйеде	в системе	жүйе	система
Жүйеге	в систему	жүйе	система
жүйелерінің	систем	жүйе	система
жүйесінің	систем	жүйе	система
мәліметтер	данные	мәлімет	данные
мәліметке	с данными	мәлімет	данные
мәліметтің	данных	мәлімет	данные
Іздеуші	поисковый	іздеуші	поисковый
іздеулерден	поиска	іздеу	поиск
Іздеуге	поиска	іздеу	поиск
бағдарлама	программа	бағдарлама	программа
бағдарламалау	программирование	бағдарламалау	программирование
бағдарламаларды	программ	бағдарлама	программа
бағдарламаға	программу	бағдарлама	программа
бағдарламасы	программы	бағдарлама	программа

We highlight two examples on the table. These are normalization of words and putting in an immutable part of the word. The first example of "ақпараттандырылған" transforms to "ақпараттандыру". Here the program cuts the suffix "ыл + ған" and adds the suffix "+ у" on the word base. Thus, we obtain the normal form of the word. Second example -tokens in most cases, which are used in the program and executed by cutting-off endings and by stemming of the word base. For example, the word " жүйелерді " takes the form of " жүйе." There are dropped the end of the right side of the case ending "- ді", then the plural "- лер".

4 Practical implementation and testing of the algorithm

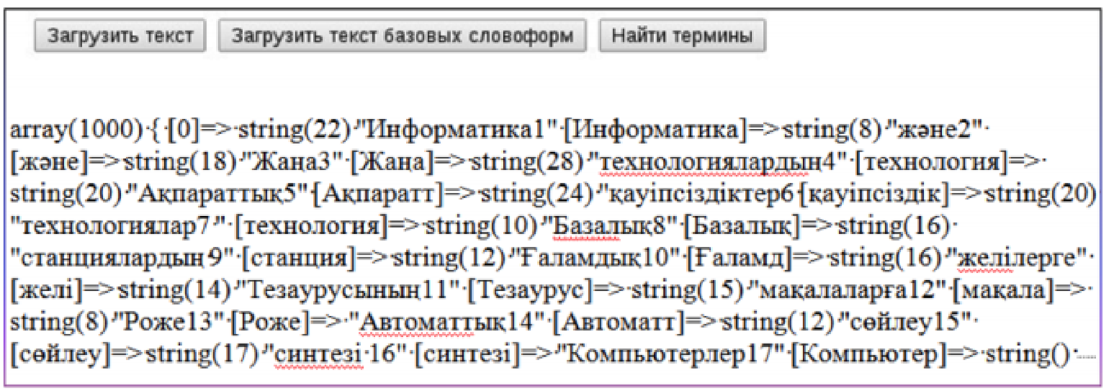
As the implementation language was chosen PHP, which is a cross-platform language, suitable for the development of embeddable into the KazTermAnalyzer module system for word processing.

There have been developed a Web application for the conversion of word forms of the Kazakh language, working with the full-text database of ICT SB RAS publications on information technology. An example application operation is shown in the picture (for the sake of brevity, showed only part of the word forms).

The checking of correctness of the word forms conversion was conducted. Initial data were taken terms from the dictionary (thesaurus), which help identify a particular word form.

As a result, for 1000 randomly selected word forms were received 100% correct generated, from what, it can be argued, that the algorithm works correctly.

As we can see, in experiments with the algorithm we get individual words in mixed form (clipped base and the normal form of the word). For example, in the word " ақпараттық " after clipping endings, remained the basis of the word "ақпаратт". And in other cases, after cutting off endings there were already normal forms of the word. For example, "компьютерлер" $\mapsto$ "компьютер", "тезаурусының" $\mapsto$"тезаурус". However, this ambiguity problem is the apparent, because it can be removed by the presence of a glossary of terms (Pic. 1).



Pic. 1. Shows the terms that have been converted into the initial form

Conclusion

The existing algorithms of separation of bases and words normalization for the Kazakh language were probed. The algorithm was developed for the morphological analyzer of the Kazakh language and the prototype of system of normalization confirming operability of this algorithm was created. In the course of work over algorithm:

– the database of initial forms of words more than 1000 words with a marking of the parts of speech and other signs necessary for generation of the dictionary of word forms was created; the formal model of word change and word formation of the Kazakh language taking into account semantics was received;

– the database of the Kazakh word forms more than 2000 dictionary entries with complete morphological information was automatically generated.

The distinctive features of created algorithm are its clearness and rather easy reproducibility that allows to apply it, in particular, without special labor costs in semantic search engines.

References

[1] Shokin Yu. I., Fedotov A. M., Barakhnin V. B., *Problemy poiska informacii [Problems of information retrieval]*, Novosibirsk, Nauka, 2010, p. 196

[2] Porter M. F. *An algorithm for suffix stripping*, Program, 1980. — T.14. — #3., p. 130–137

[3] Willett P. *The Porter stemming algorithm: then and now*, Program: Electronic Library and Information Systems, 2006. — V.3. — T.40., p. 21–223

[4] Segalovich I. V. *A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine*, 2003, p. 219–223

[5] Segalovich I. V., Maslov M. A. *Russian Morphological Analysis and Synthesis With Automatic Generation of Inflection Models For Unknown Words*, M.:Dialog, 1998. vol.2, p. 547–552

[6] *Kazakh grammar. Phonetics, vocabulary, morphology, syntax.*, Astana, Astana Polygrafia, 2002.

[7] Bektaev K. *Big Kazakh-Russian, Russian-Kazakh dictionary*, Almaty, 1995, p. 703

[8] Sharipbaev A. A., Bekmanova G. T., Ergesh B. J., Buribaeva A. K., Karabalaeva M. *Intellektual'nyj morfologicheskij analizator, osnovannyj na semanticheskih setyah [The intellectual morphological analyzer based on semantic networks].* ,Proceedings of the International Scientific-Technical Conference «Open Semantic Technology of Intelligent Systems» (OSTIS–2012). Minsk, 2012, p. 397–400.

[9] *Grammatika kazakhskogo yazyka. Sintaksis.*, Almaty: Gylym, 1967. p. 89

[10] Balakaev M. *Sovremennyi kazakhskii yazyk.*, Astana, 2006. p. 214.

A. M. Fedotov,
Institute of Computational Technologies SB RAS
Novosibirsk State University
Russia Prospect akad. Lavrent'eva, 6, Novosibirsk, 630090
Email: fedotov@sbras.ru,

J. A. Tussupov,
L. N. Gumilyov Eurasian National University, Kazakhstan Munaitpasov Street 5, Astana city, 010008
Email: tussupov@mail.ru,

M. A. Sambetbayeva,
Novosibirsk State University, Russia, Pirogova street 2, Novosibirsk, 630090
Email: madina_jgtu@mail.ru,

I. A. Idrisova,
Novosibirsk State University, Russia, Pirogova street 2, Novosibirsk, 630090
Email: indira_idrisova@mail.ru,

A. S. Yerimbetova,
Novosibirsk State University, Russia, Pirogova street 2, Novosibirsk, 630090
Email: aigerian@mail.ru.

Received 15.08.2015, Accepted 17.09.2015