Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2016"

Moscow, June 1–4, 2016

GRAMMATICAL DICTIONARY GENERATION USING MACHINE LEARNING METHODS

Mazurova M. (sleepofnodreaming12@gmail.com)

Ashmanov & Partners, Moscow, Russia

For the last decade, grammatical dictionaries have become not only a thing of theoretical value but an essential tool used in many fields of applied linguistics. However, the procedure of manual creation of a grammatical dictionary remains time- and labor-consuming. In this paper, the two-stage algorithm of automatic dictionary compilation, not requiring annotated texts, is proposed. As the source data, this system requires a formalized grammar description and a frequency distribution of a relatively large (hundred thousand tokens) corpus. Extending the principles commonly applicable to Indo-European languages, the research focuses on machine learning methods of corpora-based dictionary formation. Four machine learning models—SVM, random forest, linear regression and perceptron—are tested on the material of four languages: Albanian, Udmurt, Katharevousa, and Kazakh, and compared to a heuristic approach. While the linear models proved to be ineffective, other models' results were more promising: in an experiment with training and test sets formed from the same language's material, random forest reached 63% F-score, and SVM's results were also overdoing the baseline, however, the random forest model was unsuccessful. The best classifier in case of training and test sets based on the material of different languages was SVM. As a by-product of the experiments, the restrictions on the input were postulated: the approach 'as is' is not applicable to languages where inflections are strongly homonymic, and, on the contrary, is promising applied to an agglutinative language.

Keywords: grammatical dictionary, morphology, machine learning, morphological analyzer

ГЕНЕРАЦИЯ ГРАММАТИЧЕСКОГО СЛОВАРЯ ДЛЯ ПРОИЗВОЛЬНОГО ЯЗЫКА С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

Мазурова М. (sleepofnodreaming12@gmail.com)

«Ашманов и партнеры», Москва, Россия

Ключевые слова: грамматический словарь, морфология, морфологический анализ, машинное обучение

1. Introduction

Grammatical dictionary is a formalized description of lexemes' word formation in a language. Over the last decade grammatical dictionaries have turned out to be relevant to many tasks of modern applied linguistics. In NLP they are an essential part-of-speech (POS) taggers, spellcheckers, etc. For example, POS-tagger MyStem [12] uses a dictionary arranged as a trie of suffixes and a set of stem tries for checking if there is a canonical tagging option for a token being processed.

For Russian, there is an outstanding grammatical dictionary: 'Russian Grammar Dictionary' by A. Zaliznyak, an exhaustive inventory of Russian affixation put together in the late 1970s. It has inspired many Slavic researchers to create similar dictionaries, for example, for Polish [11] and Bulgarian [6]. As for non-Indo-European languages, there exists a Bashkir dictionary [1], but this work demonstrates a series of faults, making it difficult to use and potentially inappropriate for NLP purposes [9]. All the dictionaries listed above were compiled manually, and compilation of a grammatical dictionary is a complex time-consuming task that requires expertise of experienced linguists. In order to make the task of compiling grammatical dictionaries less problematic, an automatic approach can be suggested. In this paper I aim to develop an approach to automatic dictionary generation which does not require annotated texts and is suitable for relatively underresourced languages.

2. Related Studies

Dictionary extension methods for Russian were discussed in [13] by Segalovich & Maslov. The authors postulate a principle: 'the paradigm must be built based on corpus data'; according to it, lexeme's paradigm is a set of all its word forms found in a corpus. A program should find lexeme hypotheses using a list of suffixes and following the 'corpus-based paradigm' principle, and then filter the data with a series of heuristics. Performance analysis of the algorithm was conducted later by O. Lyashevskaya [7]. The author, however, applies another set of filters; 'the longest stem heuristics'

is proposed. The similar heuristics is used in [5] for Czech data (it is worth noting that earlier Hana & Feldman [4] used the opposite approach for a similar purpose). Later, a completely different approach to the problem is applied: for example, 'Bystroslovar' is generated from a large data array with the use of machine learning methods [14].

3. Dictionary Draft Generation

The first stage was developing a draft generation algorithm. Following the idea of the Porter's stemmer [10], the system uses a list of inflections to divide a word form into an inflection and a stem, and then combines an appropriate set of inflection-stem pairs into a lexeme as proposed in [13]. Note that a notion 'stem' (and, consequently, 'inflection') here is not linguistically correct: a stem is defined as an unchangeable part of a set of word forms of a lexeme (in the most common case, a common part of all its word forms), while an inflection is a changeable one.

3.1. Algorithm & Implementation

The system works with frequency distribution of word forms of a corpus which contains more than several hundred thousand words. The system also requires a formal description of a language's morphology in UniParser format [2].

As was mentioned earlier, the main procedure is stemming-like: the system parses a word form into a stem and an inflection, providing all parsing options allowed by the grammar used. Let us consider an example from Greek: there is an inflection '.α' in a verbal paradigm, and an inflection '*.ότερα*' in an adjective paradigm. Following the above rule, the system, given a word form '*ειδικότερα*', generates two parsing options:

(1) *ειδικότερ. + .α*
ειδικ. + .ότερα

Having produced the set of parsing hypotheses, the system attributes each of them to possible paradigm types checking which paradigm has the given affix; all the hypotheses are saved to a data storage, accompanied with a word form frequency. If there are stem alternations in the grammar, then lexeme joining is executed: the system generates alternatives for each stem found and, if there is a stem generated in the storage, puts the lexeme parts together.

The algorithm survived several different implementations [8], and the latest and the fastest one[1] is based on a custom finite-state automaton. Reading the grammar, the system forms a list including all paradigms' affixes, and then compiles an NFA capable to find multiple substrings. The NFA is used to quickly get all the parsing hypotheses, which are attributed to paradigms later by means of a structure mapping every inflection's graphical representation to a set of its attribution options.

[1] https://github.com/sleepofnodreaming/gramdicmaker2016

3.2. Performance

Below I provide a theoretical estimation of processing time. The time of NFA compilation is limited by the time necessary to read all inflections of a grammar; the compilation is executed once and its time is negligible compared to the time required for the preceding UniParser grammar compilation which is rather time-consuming.

The time required to process one word form is bounded above by an exponential function of its length because the automaton is non-deterministic; however, a possible length of a word form is limited by a small number, so the complexity may be considered asymptotically constant. As a result, the only parameter affecting the processing time is the length of an input, e.g. it is O(N), where N is a number of graphically unique word forms in the input.

Below (Table 1) I provide empirical performance measurements[2] on Kazakh data. Generally, the Kazakh corpus, 904,561 tokens in size, contains 107,704 unique word forms; for this study the word list was divided into four portions to check how the processing time increases if the input gets bigger. The number of paradigms is the same for the series of tests and equals 28.

Table 1. Performance (depending on the input frequency distribution size)

FD size, word forms	Time, s
26,926	1.855
53,852	3.523
80,778	6.797
107,704	7.005

The next question was how the processing time depends on a number of inflections in a NFA. This time, the number of words was fixed: the full frequency list was used. However, the number of affixes in an automaton varied. As adding separate inflection to the automaton is not allowed, the number of inflections in process was changed by varying the number of paradigms.

Table 2. Performance (depending on the number of inflections)

Paradigms	Number of affixes	Time, s
N-soft	766	2.01
All nominal	3,104	4.29
All verbal	82,260	5.85

[2] The machine all measurements are made on has 8Gb RAM, Intel Core i5 2,7GHz CPU, and runs MacOS 10.10.

Multiplication of a size of an input grammar does not lead to rapid growth of the processing time (see Table 2). This behavior of the system does not contradict the theoretical presuppositions made above.

4. Data Filtering

If the grammar used for the system is correct, a draft generated with the use of the described procedure includes all lexemes present in the source corpus. However, a considerable percentage of the formed lexemes are products of misparsing. For instance, example (2) was attributed to a verbal paradigm, but this set of word forms corresponds to a Kazakh noun *сұлуысың* '*beauty*':

(2) *сұлуы*.

.	*imper,2,sg / indic,prs,3*
.мыз	*indic,prs,1,pl*
.сың	*indic,prs,2,sg*

Let us call all the dictionary draft units—both real lexemes and mistakenly formed ones—pseudolexemes. In [8], a simple frequency-based heuristic classifier was used to remove false lexeme units. The threshold was set as an empirically chosen logarithmic function of a number of word forms of a lexeme: if general pseudolexeme frequency is more than a threshold value, it is considered a real one. There were also two additional thresholds: if a number of word forms of a pseudolexeme is less than the threshold-min, it is always removed; if a number of word forms is more than the threshold-max, a pseudolexeme is never removed.

Let us consider the above classifier baseline; however, in this paper I study the other approach—machine learning binary classification based on lexeme's distributional features. I also research an opportunity to use a dataset formed from another language's data.

4.1. Data Sets

For the current study I used data from the project 'Corpus Linguistics' (Udmurt, Albanian, Katharevousa) and Almaty Corpus of Kazakh[3]; besides the corpora, the source data included UniParser grammars and dictionaries[4] providing the information about lexeme's POS and paradigm type (see Table 3).

[3] http://web-corpora.net/

[4] The Katharevousa dictionary was formed automatically with the use of the frequency filter and manually reviewed. For this reason, the dictionary lacks low-frequency lexemes.

Table 3. Corpora & Dictionaries

Language	Corpus size, number of word usages	Dictionary size, lex
Katharevousa	359,805	403 (adjectives only)
Udmurt	6,368,427	21,656
Albanian	19,543,008	45,861
Kazakh	904,561	22,024/14,527

Using these sources four dictionary drafts were compiled: for Kazakh, the draft included verbal and noun pseudolexemes; for Udmurt, there were noun, verbal and adjective lexemes, and Albanian and Katharevousa draft dictionaries consisted of adjectives. Then the drafts were annotated automatically with the use of the dictionary data; no manual revision was made. Annotation was made according to the following principle: a pseudolexeme is a real lexeme if, first, all its stems postulated are a part of a lexeme found in a dictionary and, second, there is no exceeding stem in a lexeme. As a result, four data sets were formed.

Table 4. Data sets

	Lexemes formed		Valid lexemes		Invalid lexemes		Valid lexemes, %	
Language	Full	Cut	Full	Cut	Full	Cut	Full	Cut
Albanian	2,047,093	799,004	5,635	4,378	2,041,458	794,626	0.27	0.54
Kazakh	62,704	23,354	7,479	5,155	55,225	18,199	11.9	22.1
Katharevousa	3,959	—	370	—	3,589	—	9.3	—
Udmurt	278,036	101,577	7,728	5,789	270,308	95,788	2.7	5.7

Generally, a proportion of valid lexemes is not big: from language to language, it varies from 0,27% to 11,9%; but, in general, it exceeds error rate (see Table 4). However, in the case of Albanian it is abnormally low because of the language's extensive homonymy rate.

As the percent of valid lexemes turned out to be quite low and the threshold-min filter proved to be effective in [8], I formed three additional data sets, removing lexemes with frequency under 5. The thresholded data sets contain more valid lexemes, but in the case of Albanian thresholding did not solve the problem: the percentage of valid Albanian lexemes still turned out to be extremely low.

4.2. Features

The next step would be choosing a set of distributional features. Let c be a grammatical category that is represented with inflections of a paradigm π. Frequencies of c's values define probability distribution inside π: I will call it $d(c,\pi)$. Respectively, distribution

of c's values inside a pseudolexeme w is called $d_w(c,w)$. Although it is probable that distributions $d(c,\pi)$ vary dramatically from paradigm to paradigm and from language to language, it is reasonable to assume that some statistical features of a real lexeme's $d_w(c,w)$ are similar to features of $d(c,\pi)$. The set of features following the hypothesis is:

1. average entropy of $d_w(c,w)$ for the paradigm's c's, c size normalized;
2. minimum entropy of $d_w(c,w)$ for the paradigm's c's, c size normalized;
3. variance of a distribution of all entropies of $d_w(c,w)$.

Being guided by a research conducted by A. Sokirko on Russian material [14], I added a series of features based on the completeness rate of of a formed lexeme:

4. percentage of lexeme's word forms found in a corpus (graphically identical word forms are considered one form);
5. percentage of lexeme's grammatical forms found.

The next presumption is the following: if a pseudolexeme results from misparsing due to cross-paradigm homonymy, distribution of category values inside a pseudolexeme is often affected. In Katharevousa, for instance, adjective suffixes are homonymic to noun suffixes, so a noun may be interpreted as an adjective. However, an adjective pseudolexeme formed this way is going to lack the majority of word forms: a noun supposedly covers word forms of the same gender. As it is proposed in [13], these cases may be handled heuristically, but in my study I transform this idea into the following feature:

6. number of pseudolexeme's categories having the only value.

Other features that are not based on distribution of categories are:

7. entropy of word forms' frequencies, divided by a size of a paradigm;
8. word forms' frequency distribution entropy, not normalized;
9. different word form number—lexeme occurrence ratio.

5. Evaluation Methods

For evaluating the quality of a dictionary cleaning standard relevance measures were used: precision (P), recall (R) and F score. Additionally, frequency-weighted measures were set up:

$$R_f = \frac{\sum_{f\in\pi}\sum_{\omega\in tp}\sum_{s\in\omega} I(f,s)}{\sum_{f\in\pi}\sum_{\omega\in p}\sum_{s\in\omega} I(f,s)}$$

$$P_f = \frac{\sum_{f\in\pi}\sum_{\omega\in tp}\sum_{s\in\omega} I(f,s)}{\sum_{f\in\pi}\sum_{\omega\in t}\sum_{s\in\omega} I(f,s)}$$

$$F_f = \frac{2\cdot P_f \cdot R_f}{P_f + R_f}$$

The abbreviations in the formulas above are: tp—true positive results, tn—true negative results; p *и* n are numbers of positive and negative results, respectively; $I(f, s)$ is a function defining a number of tokens that may be represented as a combination of an inflection f and a stem s in a corpus.

5.1. Extension of an Existing Dictionary

In the beginning of a series of experiments, an analysis of relevance of the different features was conducted, in order to form a set of features that would be suitable for extension of an existing dictionary. Four supervised ML models were tested: SVM, linear regression, perceptron and random forest. Four training sets were used: the Kazakh and the Albanian ones, both full and thresholded. To evaluate the results, cross-validation was conducted: two-fold for Albanian case and four-fold for Kazakh case.

Table 5. Classification of Kazakh pseudolexemes

Model	Dataset	P	P_f	R	R_f	F	F_f
Perceptron	full	0.899	0.732	0.011	0.072	0.022	0.131
Perceptron	cut	0.729	0.729	0.045	0.094	0.085	0.167
Linear Regression	full	0.956	0.617	0.001	0.026	0.200	0.050
Linear Regression	cut	0	0	0	0	0	0
SVM	full	0.340	0.592	0.03	0.259	0.055	0.360
SVM	cut	0.316	0.591	0.03	0.259	0.055	0.360
Random Forest	*full*	0.540	*0.669*	*0.299*	*0.525*	*0.385*	*0.589*
Random Forest	cut	0.505	0.701	0.310	0.571	0.384	0.629
Base line	—	*0.333*	*0.410*	*0.591*	*0.939*	*0.426*	*0.571*

In this experiment (see Table 5), linear classifiers (LR and perceptron) proved to be ineffective. The performance of other classifiers is much better, but they did not outdo the heuristic one: the F score of the best ML classifier is 38.5%, while the baseline result is 42.6%. However, the weighted results are different: random forest showed to be better than all other classifiers, with F score equal to 62.9%. As for SVM, it demonstrated satisfactory weighted results, although the unweighted were poor: the weighted F score was about 36% vs. 5.5% in the unweighted case.

I will further discuss the problem with the Albanian set (see Table 6). All the classification methods used were extremely ineffective: none of the ML classifiers reached at least 6% precision, while the maximum recall was 21%. This effect can be explained by the data characteristics: the share of real lexemes is lower than the noise rate can normally be, and, as a result, the data are supposed to be not filterable with the use

of any ML. To estimate the scale of the problem, I address a specific example: a noun *ngarkesë 'load'* was attributed to each of 16 adjective paradigms specified in the grammar; although these pseudolexemes consist of one word only, a number of different grammatical forms listed is sufficient:

Table 6. Classification of Albanian pseudolexemes (cut data set)

Model	P	R	F
Perceptron	0.039	0.21	0.0658
Linear Regression	0	0	0
SVM	0	0	0
Random Forest	0.053	0.003	0.0057

(3) *ngarkes.*
.e *f.sg / f.pl / f.sg.nom.indef / f.sg.acc.indef / f.pl.nom.indef / f.pl.acc.indef*

Much more successful processing of the Kazakh data is caused by agglutinative grammatical system features: in this case, an inflection, being a combination of affixes, tends to be less homonymic. As a result, a set of grammatical categories present in a misparsed lexeme often differs dramatically from the correct one:

(4) *бағана.*
. *imper,2,sg / indic,prs,3*
.сын *imper,3*
.ғы *opt1,3*

5.2. Classification of Another Language's Data

At the next stage of the research, I tested whether it is possible to train a classifier on a data set formed from another language's material. This approach is not widespread but is used for some purposes: in [4], another language's data are used to train a HMM, and a possibility to train a morphological analyzer is postulated in [3]. As test sets, data from languages of two different types of inflection, Udmurt (agglutinative) and Katharevousa (cumulative) were used. I aimed to test the following hypothesis: it is more effective to use a training set formed of morphologically similar language's material.

Unfortunately, Kazakh turned out to be the only training set suiting for this experiment. A set of ML models was the same as in the previous experiment, but perception and linear regression's results remained extremely unsuccessful, and I will not discuss them further.

In this experiment, the use of random forest proved to be also inappropriate: in Katharevousa case, the model does not work, considering almost all the lexemes

misformed; as for Udmurt, the results are also poor: the precision is under 10%, and the recall is under 15%. However, this result is natural: the forest adapts the training data strongly, and the classification is based on certain values of a feature.

Taking a closer look on the classification results, I can see that a typical Katharevousa lexeme approved by the ML consists of a relatively big number of word forms (and, consequently, looks more like a Kazakh one). For instance, a Katharevousa lexeme *πληγή* '*wound*' (nine different forms are found) was considered a well-formed adjective, although it obviously lacks neutral and masculine forms:

(5) *πληγ.*

.άς	*pos,f,pl,acc*
.ή/.ὴ	*pos,f,sg,nom*
…	
.ῶν	*pos,pl,gen*
.ὰς	*pos,f,pl,acc*

Additionally, I studied weights of features for the classifier trained on the Kazakh data (see Fig. 1). The most significant features are #6 (a number of categories having the only value) and #8 (entropy, not normalized). The contribution of the majority of features is equally moderate (about 10%), and the only feature (minimum entropy of *(c,w)* for the paradigm's categories) proved to be useless, contributing almost nothing.

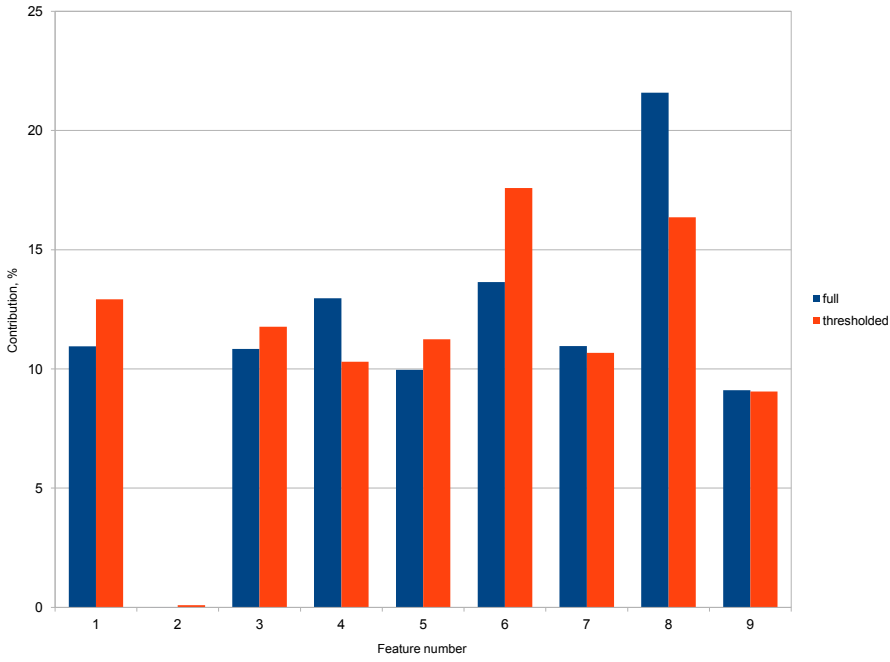

Fig. 1. Feature weights, random forest

As for SVM results, they are moderate but acceptable and still in accordance with the initial hypothesis: the precision is higher than the baseline for Udmurt, which is more similar to Kazakh morphologically. The weighted recall for Udmurt reaches 74%, and the precision is about 30%; it results in 42% F score. In the Katharevousa case the results are poorer: the maximum weighted precision and recall are 7.3% and 47.6%, respectively.

6. Conclusions

In this paper, I proposed the two-stage algorithm of grammatical dictionary generation / extension for any language. The first stage implementation, draft generation, turned out to be effective enough. As for the second stage, filtering, four ML models were tested, and two of them performed successfully. Generally, the results are moderate but promising: although the approach proposed does not work with languages featuring rich cross-paradigm homonymy, it proved to be perspective, outdoing the baseline filter both in existing dictionary extension and, conditionally, brand new dictionary generation case: it is likely that the use of the another language's training data is admissible if its inflectional model is similar to a model of a language to be processed. On the other hand, the insufficient amount usable data prevents me from conducting more detailed experiments: the dictionaries used are not completely correct from the point of view of computational linguistics, and it presumably affects the quality of results, and the number of available data sets is limited.

Acknowledgement

I am grateful to Danya Alexeyevsky, Boris Orekhov, and Andrey Kutuzov for their consultations at different stages of my research.

References

1. *Akhtyamov, M.* (1994), Bashkir Grammar Dictionary. Inflection [Башҡорт теленең грамматика һүҙлеге. Һүҙүгәреше]. 'Bashkortostan', Ufa.
2. *Arkhangelskiy, T. A.* (2012), Principles of development of a morphological analyzer for languages with different structures [Printsypy postroyeniya morfologicheskovo parser dlya raznostrukhturnykh yazykov], Moscow.
3. *Brants, T.* (2000), A statistical Part-of-Speech tagger, Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP-2000), Seattle, WA, USA, pp. 224–231.
4. *Hana, J., & Feldman, A.* (2004), Portable language technology: Russian via Czech, Proceedings of the Midwest Computational Linguistics Colloquium, Bloomington, Indiana.

5. *Kanis, J., & Müller, L.* (2005), Automatic lemmatizer construction with focus on OOV words lemmatization, Text, speech and dialogue, pp. 132–139.
6. *Koeva, S.* (1998), Bulgarian Grammatical dictionary. Organization of the language data, Bulgarian language, Vol. 6, pp. 49–58.
7. *Lyashevskaya O. N.* (2007), Towards the lemmatization of word forms absent from dictionary [K probleme lemmatizatsii neslovarnykh slovoform], Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2007" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2007"], Bekasovo, pp. 407–412.
8. *Mazurova, M.* (2014), Generating a farmalized description of a language's lexis from unannotated texts [Porozhdeniye formalizovannogo opisaniya leksiki yazyka na osnove tekstov], Moscow.
9. *Orekhov, B. V.* (2014), Problems of grammatical annotation of texts in Bashkir [Problemy morfologicheskoy ravmetki bashkirskih tekstov], Proceedings of TEL-2014 [Trudy Kazanskoy shkoly po kompjuternoy i kognitivnoy lingvistike TEL-2014], Kazan, pp. 135–140.
10. *Porter, M. F.* (1980), An algorithm for suffix stripping, Program, Vol. 14(3), pp. 130–137.
11. *Saloni, Z., Gruszczyński, W., Woliński, M., & Wołosz, R.* (2007), Grammatical Dictionary of Polish, Studies in Polish Linguistics, Vol. 4, pp. 5–25.
12. *Segalovich, I.* (2003), A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine, MLMTA, Los Angeles, pp. 273–280.
13. *Segalovich, I., Maslov, M.* (1998), Russian morphological analysis and synthesis, generating inflectional models for word absent from dictionary [Russkiy morfologicheskiy analiz i sintez s generatsiyey modeley slovoizmeneniya dlya ne opisannykh v slovare slov]. Dialogue'98 [Dialog 98], Kazan, Vol. 2, pp. 547–552.
14. *Sokirko A. V.* (2010), Bystroslovar': morphological prediction of new Russian words using very large corpora [Bystroslovar': predskazanie morfologii russkikh slov s ispol'zovaniem bolshikh lingvisticheskikh resursov], Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2010" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2010"], Bekasovo, pp. 450–456.
15. *Zaliznyak A.* (1977), Russian Grammar Dictionary [Grammaticheskij slovar' russkogo jazyka], Moskva.