Polyphon: An Algorithm for Phonetic String Matching in Russian Language

Viacheslav V. Paramonov(✉), Alexey O. Shigarov,
Gennagy M. Ruzhnikov, and Polina V. Belykh

Matrosov Institute for System Dynamics and Control Theory of Siberian Branch
of Russian, Academy of Sciences (ISDCT SB RAS), Irkutsk, Russia
{slv, shigarov, rugnikov, polina}@icc.ru

Abstract. Data cleansing is the crucial matter in business intelligence. We propose a new phonetic algorithm to string matching in Russian language without transliteration from Cyrillic to Latin characters. It is based on the rules of sounds formation in Russian language. Additionally, we consider an extended algorithm for matching of Cyrillic strings where phonetic code letters are presented as primes, and the code of a string is the sum of these numbers. Experimental results show that our algorithms allow accurately matching phonetically similar strings in Russian language.

Keywords: Phonetic algorithms · String matching · Language · Classifiers

1 Introduction

Often, data integration as an important part of the business intelligence encounter messy data, including typos, misspelling strings, and repeated words. These problems require preliminary data cleansing which consists of many aspects such as detection and correction of spelling errors, missing data, incorrect values, and logical inconsistencies [1, 2]. One of the important tasks of data cleansing is to associate source values (natural language strings) with target thesauruses and qualifiers. In this task, phonetic algorithms [3] can be used to match phonetically similar strings.

The coding of strings (words) in phonetic algorithms is based on their pronunciation, but not on their spelling. Substantially, the familiar phonetic algorithms (e.g. Soundex, Metaphone, Double Metaphone, and Caverphone) are intended to find and match specific names which are written in Latin. In the case of text processing in the languages with rich morphology, e.g. Russian, there is a need to adapt phonetic algorithms to their features [4].

This paper introduces the method of detection and correction of Russian language spelling errors in data normalization processes. The proposed algorithm is based on matching analysis of phonetic coded strings.

Russian language belongs to East-Slavic language group. It is used by more than 250 million speakers [5]. The main aim of this research is to develop an algorithm of phonetic coding of words for Russian language.

G. Dregvaite and R. Damasevicius (Eds.): ICIST 2016, CCIS 639, pp. 568–579, 2016.
DOI: 10.1007/978-3-319-46254-7_46

A phonetic algorithm seeks to identify words according to similarities in their pronunciation [6]. The common purpose of phonetic algorithms is detection of the words similarity according to their phonetic resemblance. The most typical application of phonetic algorithms is intended for the surnames comparison [4].

Most of the existing phonetic algorithms are developed for English language [6]. Some of them support other languages partially [4, 6]. However, in our literature review, we were not able to find any research concerning with phonetic algorithms for Russian language where Cyrillic symbols are used. Implementation of phonetic algorithms for languages of this kind is usually implemented by transliterating Cyrillic characters to Latin. This approach lead to ignoring language phonetic features [4]. The aim of the research is to develop a phonetic algorithm that takes into accounts the rules of phonetic coding in the Russian language and analyses their applicability to the Russian language and possibility of their extending to other languages of East-Slavic language group such as Belorussian and Ukrainian.

Phonetic coded string matching is useful for eliminating errors occurring in typing processes of text values that might be compared with classifiers. The examples of qualifiers are KLADR (Russia' address database) [7], IPNI (International Plant Names Index) [8], ICD X (International Statistical Classification of Diseases and Related Health Problems 10th Revision) [9] etc.

It worth mentioning that data input is often accompanied by spelling errors occurring in words. It is done unintentionally in most cases but it leads to data incorrectness and complicates further processing. The rate errors in data entry process by an operator is about 5 % [10].

The first part of this paper considers phonetic spelling errors. The second part introduces phonetic algorithms for data cleansing tasks. Thirdly, the a phonetic algorithm for Russian language, Polyphon, is described. The last part describes data preparation for the algorithm testing and test results.

2 Phonetic Algorithms for Data Cleansing Tasks

2.1 Phonetic Spelling Errors

Spelling errors in Russian language words could be classified in following groups [11]:

- morphological – a uniform graphic symbol of morphemes by the letter i.e. a person tries to write all audible sounds by letters [12];
- phonologic – preservation of writing of phonemes spelling regardless the word of change;
- phonetic – words written as they are heard;
- traditional (historic) – writing by "tradition" i.e. as it was written in old times or as in the language which it is borrowed from.

Most of spelling errors in Russian are associated with phonetic norms. The type of mistakes generally depends on the level of one's education [13]. For example, elementary school students make mistakes because they write words as heard. High school

students are more prone to hypercorrection errors. As the literacy level of a person does not improve after school graduation, adult persons also make hypercorrection errors. However, in general, many spelling mistakes are simply typos or associated with phonetic words representation.

Table 1. Share of most various spelling patterns in total number of mistakes

Place	Spelling pattern type	Share (in %)
1	Mixing of "**и**"-"**е**" in unstressed syllable	27.3
2	Mixing "**а**"-"**о**" in unstressed syllable	25.3
3	Separate writing instead of solid word writing	9.4
4	Solid word writing instead of the separate	8.8
5	Writing of one letter instead of the doubled	6.6
6	Mixture deaf and ringing letters	3.6
7	Vowels after hissing sounds and ц	2.7
8	Excess doubling of letters	2.6
9	Absence of **ь** character	1.3
10	Writing of the superfluous **ь** character	0.6

It is noted in [13] that phonetic spelling principle is the most natural and efficient. Therefore phonetic algorithms would be useful for similar strings matching. Table 1 represents Russian language general spelling errors. It is a simplified table from [13]. Thus, these kinds of spelling errors make nearly 90 % of all errors. Other spelling errors are represented by mixing of some vowels, concordant, letter sequences or dividing (special) letters "**ь**" and "**ъ**". Forming rules that allow to find words matching with due regard to errors type help to find and eliminate errors in words.

Here and in other examples Russian language words will be presented in transliteration according to [14] and shown in square brackets "**[]**" characters. This notation is aimed to help understanding of text for persons who are not familiar with the Cyrillic characters.

2.2 Textual Data Cleansing

We typically deal with nouns and adjectives in singular and nominative form in data cleansing. Therefore, it is reasonable to apply fuzzy string comparison methods and phonetic algorithms. Fuzzy string comparison algorithms aid to determine level of string similarity. Phonetic algorithms help to clarify identity of similar words.

(1) **Fuzzy string comparison**

We assume that the first stage of data cleansing in case of qualifiers linked data is a fuzzy string comparison. The duplicating values (words) in lexeme are not considered. There are many lexemes processes realization for East-Slavic group languages. Samples of words normalization are shown in Table 2.

Table 2. Lexeme words normalization sample

Lexemes	Words in normalized form	Language
General Dovator streets	"dovat" "general" "street"	English
улицы Генерала Доватора [ulitsy Generala Dovatora]	генера доватор улиц [genera dovator ulits]	Russian
вулиці генерала Доватора	вулиц генера доватор [vulits genera dovator]	Ukrainian
вуліцы генерала Даватара	вулиц генера даватар [vulits genera dovator]	Belorussian

Algorithms for fuzzy string comparison are applied to estimate similarity of normalized lexemes. If the measure of similarity with predetermined values by operator is not 0, then phonetic algorithms are applied to achieve the precise result.

One of the popular methods of substring search in the text is fuzzy string search. These kind of algorithms use for orthography check and for text search. The well-known search engines as Yandex (http://yandex.com) and Google (http://google.com) both use fuzzy search algorithms [15, 16]. The common idea of these algorithms is as following: by a given word it is possible to find in a text or in a dictionary all words which is matching the given word (or starting with it) in view of k possible differences.

Fuzzy string comparison is used in word and expression search. This kind of algorithms is language independent (e.g. with non-Hieroglyphic characters).

Phonetic Algorithms. Phonetic algorithms are used for estimation of word similarity according to their phonetic forms. This coding depends on pronunciation particularities but not on orthographic rules. The word is phonetically similar to the matching codes.

This kind of algorithms allow to find typos related to changing places of two adjacent letters and typos based on sound similarities. We suggest that the following categories of errors are allocated to estimate overall performance of phonetic algorithms:

- First category – errors associated with incorrect writing (morphological errors).
- Second category – various typographical errors (misprints, typos).
- Third category – errors related to incorrect rule usage (hypercorrection [13] errors).

There are many phonetic algorithms, which use Latin characters or English language rules and some local dialects of it while particularities of Cyrillic letters in East-Slavic group languages and their sounds are not taken into account. Using well-known phonetic algorithms for Russian language texts usually results in transliteration of Cyrillic letters to Latin. Transliterated words do not always have an unique record. For example, such name as "АНТОН ЧЕХОВ" **["ANTON CHEKHOV"]** has been variously transliterated as TSJECHOF, TSJECHOW, TJEKHOW, CHEKHOV, CHEKHOW etc. Further addition of the word suffix will complicate transliteration.

Appling phonetic codes allow increasing the word comparison quality in the case of the incorrect writing [17]. All phonetic algorithms use words coding. Changing of noun case and form, for example, leads to ineffective use of phonetic algorithms. However, such changes are not significant in our case. Therefore, phonetic algorithms are most

suitable for word comparison with reference books or dictionaries. Phonetic algorithms are used expediently for the solution of issues of comparison word with their meanings in reference books or dictionaries (classifiers).

3 Phonetic Algorithms for Russian Language

3.1 Polyphon: Russian Language Phonetic Algorithms Adaptation

The common phonetic algorithms originally are intended for English language. They can be applied to non-Latin characters after transliteration, e.g. from Cyrillic letters to Latin ones. There are many ways to transliterate some letters. For example, we can transliterate the Ukrainian word « вчора » (yesterday) as "vchera", "vchora", "fchora". In addition, misspellings in East Slavic languages with Cyrillic letters generally differ from these in English or German texts due to dissimilar rules of pronunciation and writing in different languages. However, it is unable to consider language phonetic features of letter sequences for each language in transliteration. Thus, the most known phonetic algorithms are not so effective for texts with non-Latin characters.

The proposed algorithm Polyphon uses word transformation according rules of Russian language and its phonetic particularities. It transforms words to codes as well as others phonetic algorithms. The codes matching define words phonetic similarity. The algorithm allows to get a more accurate phonetic code for conformable strings according phonetic rules of Russian language. The stages of algorithm are:

- substitution of Latin letters which are similar to Russian with Russian letters;
- removal of all non-Russian characters from the string;
- removal dividers (special characters) from string;
- transformation of doubled characters into one;
- transformation of character sequences.

Details of Polyphon Algorithm. Some letters in Russian alphabet have equivalent in writing with Latin. These are letters: a **[a]** ~ a, e **[e]** ~ e, o **[o]** ~ o, c **[es]** ~ c, x **[kha]** ~ x. Some letters equal in capital letters only: B **[ve]** ~ B, M **[em]** ~ M, H **[en]** ~ H. Sometimes these letters are substituted (incidentally or purposely) when text typing. Thereunder such Latin characters replace by corresponding Russian. It is preparatory stage of the algorithm.

There are some dividers – special letters "**ь**" ("soft" sign), "**ъ**" ("hard" sing) presented in Russian language. They are not pronounced and used for giving softness or hardness for consonants respectively. For this reason, there is no need to consider these characters.

The developed phonetic algorithm Polyphon operates with Russian language letters only. The initial operation is to remove all characters, which are not in the Russian alphabet.

The following stage is transformation of letters repeated in a row. Doubled letters will transform to one e.g. "xx" to "x". It is not always possible to define double letters in hearing. Consequently, we carry out these transformations for rule of generalization.

Table 3. Processing a set of words

Standard value	Code	Fuzzy phonetic equivalents
генерала Доватора [GENERALA DOVATORA]	154	Доватора генерала [DOVATORA GENERALA] генерала Доватора [GENERALA DOVATORA] генирала даватора [GENIRALA DOVATORA]

Table 4. Replacement of some letters

Letters	А, Е, Ё, И,О, Ы, Э, Я	Б	В	Г	Д	З	Щ	Ж	М	Ю
Modification result	А	П	Ф	К	Т	С	Ш	Ш	Н	У

Table 5. Letters sequence conversion

Sequence	АКА	АН	ЗЧ	ЛНЦ	ЛФСТФ	НАТ	НТЦ	НТ
result	АФА	Н	Ш	НЦ	ЛСТФ	Н	НЦ	Н
Sequence	НТА	НТК	НТС	НТСК	НТШ	ОКО	ПАЛ	РТЧ
result	НА	НК	НС	НСК	НШ	ОФО	ПЛ	РЧ
Sequence	РТЦ	СП	ТСЯ	СТЛ	СТН	СЧ	СШ	ТАТ
Result	РЦ	СФ	Ц	СЛ	СН	Ш	Ш	Т
Sequence	ТСА	ТАФ	ТС	ТЦ	ТЧ	ФАК	ФСТФ	ШЧ
Result	Ц	ТФ	ТЦ	Ц	Ч	ФК	СТФ	Ш

It is taken into account that phonetic code depends on the sound that can correspond with some letters or their sequences. Some different letters or their combinations have different sounds. Accordingly, Polyphon codes letters by sounds as heard. The ways of the replacement are provided in Table 3. The aim of the proposed phonetic algorithm is to generalize letters and sounds combinations [18]. The reason of generalization is based on idea that some sounds form letter sequences according their stress position. Such deviations from norms are common in social and territorial dialects in Russia [19].

There is a reduction of vowels occurring in Russian language when a word has 3 or more syllables. Vowels at the beginning and the end of the word are remained. Therefore in a word which contains 3 and more syllables the algorithm will remove all vowels in the middle of the word. The basis of splitting a word into syllables is the number of vowels in the word. Some vowels may be placed in the word with an error. We assume that if there are more than 4 consonants they will form 2 syllables.

The next stage is the substitution of the sequences of letters taking into account the changes made in Table 4. Examples of letters sequence substitution are presented in

Table 5. Often a combination of letters leads to different sound. The data from [18] was used as a basis for these combinations.

The result of word transformation is a phonetic code where consecutive same letters are replaced, for example: “телегаммааппарат” [**telegrammaapparat**] – “тэлэгамаапарат” [**telegramaaparat**].

We extended the Polyphon application to using phonetic code as primes. Firstly, all repeated letters are deleted from the string. Therefore, one letter presented in the string one time only. Each letter has a prime numerical code according to Table 6. The resulting code is the sum of primes. Usage of the sum of primes guarantees that strings with different letters will have different codes. This algorithm extension uses fuzzy phonetic comparison.

3.2 Fuzzy Phonetic Comparison

The word represented in phonetic code can be shown as a sum of primes. In this way, it is possible to use fuzzy phonetic comparison to extend the area of algorithm applicability. The algorithm facilities process phrase treatment when the word order can be broken. The example of such phrases is given in Table 6. It is possible to eliminate typos related to shift of letters also. Example of these typos is “компьютер” [**comp’yuter**] –“компьюетр” [**comp’yuetr**].

It is necessary to remove all duplicating letters from phrase except one code word by primes. The resulting code of the word will be the sum of letter coding. If the resulting code of two words is identical to the meaning, so the words are phonetically similar.

The essence of the algorithm is modification of words, processing a certain number of letters and summation of their codes. The algorithm for phonetic words coding is offered that considers phonetic particularities of Russian language.

Table 6. Letters coding

letter	а	п	к	л	м	н	р	с	т
code	2	3	5	7	11	13	17	19	23
letter	у	ф	х	ц	ч	щ	э	я	
code	29	31	37	41	43	47	53	59	

4 Experimental Testing of Polyphon Algorithm

4.1 Description of Experiment

We perform the following experiment as to efficiency and accuracy of the proposed algorithm:

The experiment consists of several stages:

- data preparation: to generate a data set of words with mistakes;
- testing of phonetic algorithms;
- comparison with existing algorithms.

The basis for testing – words from Ozhegov' explanatory dictionary [20]. The words without their description were used for the experiment. Some words are identical because a word could have more than one meaning. The initial amount of words for error introduction is 11601.

The method for errors generation was developed. The errors, which expressed in words, reflect the phonetic phenomena and processes of Russian language. First category errors consider such processes as:

- **position changes** – the phonetic rule at the end of the word (devocalization of a paired consonant on the end of the word) and reduction (a qualitative reduction is letters substitution e.g. "о" **[o]** and "и" **[i]**, "е" **[e]** to "и" **[i]**, etc. in a weak position).

  We used a positional stunning and voicing of consonants. Voiced pair stunned at the end of words and before voiceless consonants. (мозг {ск}**[mozg {sk}]**, параход {т} **[parahod {t}]**). Voiceless consonants converts to voiced consonants every time, in the case when their location before voiced (сдать {здать} **[sdat' {zdat'}]**). The exception is the unpaired voiced consonants and "в" character. In the diaeresis process one sound is removed out and a different sound appears (сердце {с'эрцъ} **[serdtse {s'ertc'}]**, солнце {сонцэ} **[solntse {solntce}]**). The fusion process is merging of consonants (жарится моется **[zharitsya – moetsya]** – жарит(ц)а **[zharit(c)ya]**, мыться **[myt'sya]** – мы(ц)а **[my(tc)a]**).
- **assimilation and dissimilation processes** - devocalization and vocalization of concordats in the word. The phenomenon of assimilation is the similarity of sounds, i.e. (ножка {шк} **[nozhka {shk}]**, отдать {дд} **[otdat' {dd}]**, сдоба {зд} **[sdoba {zd}]**, косьба {зьб} **[kos'ba {z'b}]**).

  The basis for this category of errors is Russian language it is orthography errors in unstressed vowels and "ъ", "ъ".

  Wrong writing of "ь" for assimilation softness of consonants in combinations зд(*) **[zd]**, -ст(*) **[st]**,-зн(*) **[zn]**,-тн(*) **[tn]**,-сн(*) **[ch]**,-ст(*) **[st]**,-нн(*) **[nn]**, -нч(*) **[nch]**, -нщ(*) **[nsh]**, - нт(*) **[nt]**, -дн(*) **[dn]**, where (*) is vowel е **[e]**, ё **[jo]**, ю **[yu]**, я **[ya]**, и **[i]**. (гвозьди **[gvozdi]**, есьть **[es't']**, жизьнь **[zhizh'n']**, защитьник **[zash'itnik]**, лисьтья **[list'ya]**, раньнего **[ran'ego]**, сеньтябрь **[sentyabr']**, утреньнюю **[utren'yuyu]**, шерсьть **[shers't']**, коньчились **[konchilis']**, опусьтели **[opus'teli]**, отьнес **[otnes]**, песьня **[pes'nya]**, полдьню **[pold'nyu]**). The submission of "ъ" instead of "ъ" in words with "ь" before vowels "е", "ё", "ю", "я", "и". (бьют **[b'yut]** – бъют **[byut]**) and (сьезд **[s'ezd]**) on the contrary was considered as well.

  Errors of incorrect « не » **[ne]** and « ни » **[ni]** writing were not considered. Errors of letters mixing and their shift are not generated.

The software use all words and tries entering errors of each type into the word if it is possible. The number of generated words, which contain errors, is 50196.

The resulting document has two columns – original "correct" word and the same word with phonetic error(s). Examples of words with mistakes are shown in Table 7.

Table 7. Example of words with mistakes

Original word	Word with mistak(es)
АВАНЗАЛ **[AVANZAL]**	АВВАНЗАЛ **[AVVANZAL]**
	АВАНЗЗАЛ **[AVANZZAL]**
	ЕВАНЗАЛ **[EVANZAL]**
	АВАНЗАЛЛ **[AVANZALL]**
	АВААНЗАЛ **[AVAANZAL]**
	АВАНЗАЛ **[AVANZAL]**
	АВАННЗАЛ **[AVANNZAL]**
	ААВАНЗАЛ **[AAVANZAL]**
	АВАНЗААЛ **[AVANZAAL]**

Table 8. Algorithms comparison results

Algorithm	Matches of phonetic codes (in %)	Time (in milliseconds)
Proposed algorithm	95.12	2003
Proposed algorithm (fuzzy phonetic comparison)	98.8	1623
Soundex	90.24	1096
Metaphone	90.29	870
Double Metaphone	96.15	1451
Caverphone	90.41	9770
NYSIIS	75.97	1517
DaitchMokotoffSoundex	96.84 %	1763

4.2 Algorithm Testing and Comparison

The proposed algorithm is applied to a prepared set of test data. As a result we have an accurate verification of all words from a reference. We compare such phonetic algorithms as Soundex, Metaphone, Caverphone, Daitch-Mokotoff Soundex (implemented in Java language package org.apache.commons.codec.language). These algorithms use English alphabet characters only as in the Russian standard of transliteration GOST R 52535.1-2006 [14].

The results of testing is obtained by the Polyphon algorithm, Double Metaphone, Caverphone and Daitch-Mokotoff Soundex are shown in Table 8. Note that strings which were shown as different in Double Metaphone, Caverphone and Daitch-Mokotoff Soundex are displayed as equal in the proposed algorithm. Moreover, the algorithm has been tested on single words only.

A part of information about a word might be lost by any phonetic algorithm which can lead to wrong comparisons. Word transformation with the suggested approach allows comparing word according to their possible phonetic transformation. The suggested approach permits to compare words accurately. Unrecognized words represent words with several types of mistakes, including reduction.

Table 9. The results of estimating the accuracy for the word matching.

Range of code coincidences	Number of ambiguous cases
More than 100	0
from 10 to 100	608
from 5 to 9	2056
from 3 to 4	10034
1	37496
0	2

We also estimate the accuracy for the word matching as follows. We match in pairs each misspelled word with each reference word from Ozhegov' dictionary. If their codes are identical then we increment the number of code coincidences with the misspelled word. The experimental results are shown in Table 9. For example, the misspelled word "литие" **[litie]** and the five reference words "ладо" **[lado]**, "литье" **[lit'e]**, "летие" **[letie]**, "лето" **[leto]**, and "леди" **[ledi]** have the same code "лата" **[lata]**. It means that we have one ambiguous case of the word matching in the range from 5 to 9. The experimental results demonstrate the high rate of accuracy for the word matching.

5 Conclusions

In this paper, we presented the algorithm for phonetic word comparison including fuzzy phonetic comparison option. This algorithm can be used not only for surnames but also for establishing corresponds of the word meaning to the qualifier entry. The described approach is useful for data integration process. The proposed methods can be applied into data cleaning tools for Russian text processing. Accurate data clean-up is of help for integrating data from different sources.

The proposed approach is based on Russian language phonetic rules. Phonetic coding is more exact in comparison with the algorithms based on transliteration. We suggest that our algorithm could be used not only for surnames but for establishing compliance of word meaning to qualifiers. Letters transformation rules allow it to be used for languages similar to the Russian language, such as Belorussian, Ukrainian.

Phonetic fuzzy string coding allows to process a large number of similar words. As for further work, in order to improve the effectiveness of the proposed algorithm, fuzzy string comparison methods need to be included.

Acknowledgments. The reported study was supported in part by RFBR (grants 15-37-20042, 15-47-04348, 16-07-00411, and 16-57-44034); Council for Grants of the President of Russian Foundation (grant NSh-8081.2016.9). Experiments were performed on the resources of the Shared Equipment Centre of Integrated information and computing network of Irkutsk Research and Educational Complex (http://net.icc.ru).

References

1. Müller, H., Freytag, J.-Ch.: Problems, Methods, and Challenges in Comprehensive Data Cleansing, pp 5–12. Berlin University (2003)
2. Maletic, J., Marcus, A.: DataCleansing: A Prelude to Knowledge Discovery. Data Mining and Knowledge Discovery Handbook, pp. 19–32. Springer, Heidelberg (2010)
3. Zobel, J., Dart, Ph: Finding approximate matches in large lexicons. Softw. Pract. Exp. **25**(3), 331–345 (1995)
4. Zahoransky, D., Polasek I.: Text search of surnames in some slavic and other morphologically rich languages using rule based phonetic algorithms. In: Processing, IEEE/ACM Trans on Audio, Speech, and Language (T-ASL), pp. 553–563. IEEE (2015)
5. Cubberley, P.: Russian A Linguistic Introduction, p. 369. Cambridge press, New York (2002)
6. Parmar, V.P., Kumbharana, C.K.: Study existing various phonetic algorithms and designing and development of a working model for the new developed algorithm and comparison by implementing it with existing algorithm(s). Int. J. Comput. Appl. **98**(19), 45–49 (2014). (0975 – 8887)
7. Russia' address classifiactior. Tax Service of Russia (Классификатор адресов России (КЛАДР)). http://www.gnivc.ru/inf_provision/classifiers_reference/kladr/ (in Russian)
8. The International Plant Names Index (IPNI). http://www.ipni.org/
9. International Statistical Classification of Diseases and Related Health Problems 10th Revision. http://apps.who.int/classifications/icd10/browse/2016/en
10. Orr, K.: Data quality and systems theory. Commun. ACM **41**(2), 66–71 (1998)
11. Skripnik, Ya.N., Smolenskaya, T.M.: Phonetic of modern Russian language: study book. (Скрипник Я.Н., Смоленская Т.М. Фонетика современного русского языка: Учебное пособие / Под ред. Я.Н. Скрипник.) Stavropol, 152p (2010). (in Russian)
12. Valgina, N.S., Rozental, D.E., Fomina M.I.: Modern Russian language: Textbook (Валгина Н.С., Розенталь Д.Э., Фомина М.И. Современный русский язык: Учебник) 6th edition Moscow: Logos. 2002 – 528 p (in Russian)
13. Osipov, B.I., Galushinskaya, L.G., Popkov, V.V.: Phonetic and hypercorrection errors in written assignments of pupils of 3-11 classes of high school (Фонетические и гиперические ошибки в письменных работах учащихся 3–11-х классов средней школы). Russian Language journal. # 15, 2002 (in Russian). http://rus.1september.ru/article.php?ID=200201501
14. GOST R 52535.1-2006. Identification cards. Machine readable travel documents. Part 1 Machine Readable Passports. National Standard of the Russian Federation (ГОСТ Р 52535.1-2006. Карты идентификационные. Машиносчитываемые дорожные документы. Часть 1. Машиносчитываемые паспорта. Национальный стандарт Российской Федерации). Moscow, Russia, 18 p (2006). (in Russian)
15. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Comput. Netw. ISDN Syst. **30**(1), 107–117 (1998)
16. Haveliwala, T.: Efficient computation of pagerank. Technical Report 1999-31. Stanford University (1999). http://dbpubs.stanford.edu/pub/1999-31
17. The Soundex Indexing System. National archives. http://www.archives.gov/research/census/soundex.html
18. Ivanova, T.F.: New orthoepic dictionary of Russian. Pronunciation. Accent. Grammatical forms (Иванова Т.Ф. Новый орфоэпический словарь русского языка. Произношение. Ударение. Грамматические формы) Second edititon. – Russian language-Media, 893 p. (2005) (in Russian)

19. Zhirmunsky, V.: National Language and social dialects (Жирмкнский В. Национальный язык и социальные диалекты).Moscow: The state publisher of fiction, 300 p. (1936). (in Russian)
20. Ozhegov, S.I.: Dictionary of Russian language. About 53000 words. (Словарь русского языка: Ок. 53 000 слов) / Editor Skvortsova L.I. Edition 24, Moscow: Oniks, World and education, 1200 p. (2007). (in Russian)