(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 July 2011 (21.07.2011)

PCT

(10) International Publication Number
WO 2011/087391 A1

(51) International Patent Classification:
G06F 17/28 (2006.01)

(21) International Application Number:
PCT/RU2010/000013

(22) International Filing Date:
18 January 2010 (18.01.2010)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **GOOGLE INC.** [US/US]; 1600 Amphitheater Parkway, Mountain View, California, (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **USHAKOV, Maxim Aleksandrovich** [RU/RU]; ul. Leskova, d. 9, kv. 188, Moscow, 127349 (RU). **GLOTOV, Denis Yurievich** [RU/RU]; ul. Pivchenkova, d. 8, kv. 57, Moscow, 121108 (RU).

(74) Agent: **DEMENTIEV, Vladimir Nikolaevich**; Gowlings International Inc., Moscow Representative Office, Prechistensky Pereulok 14, Bldg. 1, 4th Floor, Moscow, 119034 (RU).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

WO 2011/087391 A1

(54) Title: AUTOMATIC TRANSLITERATION OF A RECORD IN A FIRST LANGUAGE TO A WORD IN A SECOND LANGUAGE

(57) Abstract: This specification describes an innovative method for automatic transliteration of a record in a first language to a word in a second language.

AUTOMATIC TRANSLITERATION OF A RECORD IN A FIRST LANGUAGE TO A WORD IN A SECOND LANGUAGE

Technical Field

The present specification relates to a method for automatic transliteration of records in a first language to relevant words in a second language.

Background Art

Transliteration is used in many modern applications and particularly in text applications for mobile telephones, e.g., SMS (Short Message Service), and for network data retrieval, e.g., in Internet search engines.

As a rule, transliteration, i.e. recording the words of one language by means of characters of the other language, is carried out with use of tables of correspondence between letters or more often between sounds, sometimes between syllables in a first language and their representation by letters of a second language. Such methods are described, for example in the US Patents 6460015 (published on 01.10.2002), 6810374 (published on 26.10.2004), 7376648 (published on 20.05.2008), in US Patent Applications 2005/0182616 (published on 18.08.2005), 2005/0216253 (published on 29.09.2005), 2006/0143207 (published on 29.06.2006), 2007/0288230 (published on 13.12.2007), 2008/0097745 (published on 24.04.2008), as well as in International Patent Application WO 01/20435 (published on 22.03.2001). A disadvantage of all such methods is failure to unambiguously transliterate the same combinations of letters or sounds in various words.

Chinese Patent 1193780 (published on 23.09.1998) describes a "multi-purpose" method of transliteration. Such multi-purpose nature consists in the fact that initially any language would be translated into Esperanto, and afterwards transliterated with Latin letters. It is clear that the extra stage of

representing the text in Esperanto due to obligatory participation of a human would inevitably lengthen the transliteration process, making it more expensive, and introduce extra errors.

The US Patent Application 2008/0270111 (published on 30.10.2008) describes a transliteration method, wherein each vowel and consonant has unique representation in Unicode. In this method various phonetic and pseudo-phonetic transliteration variants are embodied, and generated words with preset information about them are grouped. The resultant variants are analyzed with due consideration for such information, applying preset transliteration rules. A disadvantage of this method lies in that letters are uniquely represented in code equivalent, since various combinations of letters may sound differently, which is not taken into account.

Canadian Patent Application 2630949 (published on 21.11.2008) describes a transliteration method that involves an attempt of replacing chains of letters in a first language with chains of letters in a second language, determination of the replacement probability, and then selection of the most probable replacement based on a preset criterion. A similar method is used in the laid-open Japanese Patent Application 2005-092682 (published on 07.04.2005). However, both methods are rigidly linked to the pair of languages considered in each method (English and Arabic or English and Japanese, respectively) that are poorly flectional. Therefore, they are hardly adaptable to other cases.

Summary

This specification describes an innovative method for automatic transliteration of a record in a first language to a word in a second language. This method includes the following: receiving a sequence of signals, each of which encodes the corresponding character in a first language record to be transliterated, and saving the received signal sequence to a memory;

during stepwise analysis of the signals from the start to the end of the saved signal sequence, finding all transliteration rules for the character being analyzed at the given step or character group starting with the character being analyzed at the given step, in the first language record to be transliterated, by looking up in an appropriate rule base created in advance;

complementing all the transliteration variants obtained at the previous stage with characters of the second language from each transliteration rule found at the given step, thus obtaining at the given step composite variants of transliteration associated with this step or with one of the subsequent steps;

at each step, comparing each of the composite transliteration variants obtained at the given step against starting segments of words in the second language, by looking up in an appropriate base of starting segments of words in the second language, and if a particular transliteration variant associated with the given step matches a starting segment of a word in the second language, retaining this transliteration variant for analysis in subsequent steps;

on completion of the last step of the analysis, accepting at least one transliteration variant in the second language associated with the last step as the transliteration result of the record in the first language; and

creating a sequence of signals, each of which encodes the corresponding character in the resulting transliterated word of the second language.

An optional feature of the method is that the received signal sequence is complemented at the start and end with signals corresponding to preset characters, and the first step of analysis is started from the first of the preset characters. The preset characters can be alphabetic or non-alphabetic, and they may be different characters or the very same preset character.

Another optional feature of the method is that the characters used for the record in the first language are selected from the group made up of: alphabetic characters of the first language; non-alphabetic characters, each resembling

some alphabetic character of the second language; and non-alphabetic characters that in combination resemble some alphabetic character of the second language.

One more optional feature of the method is that the base of starting segments of words in the second language is created directly during the comparison, from chains no longer than m characters, which chains are obtained from actual words of the second language, and the starting segments of words in the second language are created by imposition of the chains with a shift by one character, wherein $m > 1$ is an integer pre-selected for the second language; during this process, the occurrence frequency in the second language is determined for each of the chains, and the chains with occurrence frequency below a preset limit are removed.

In the latter case, for each particular transliteration variant associated with the given step and no more than m characters long, its probability is determined by finding the occurrence frequency of the corresponding chain, while for each particular variant that is longer than m characters and is associated with the given step, its probability is determined by multiplying the occurrence frequencies of those involved chains m characters long which belong to the given particular transliteration variant with overlapping upon shifting each subsequent chain m characters long by one character from the start of the given particular transliteration variant; if any of the chains is missing, the variant is discarded.

At the same time, during determination of the probabilities of transliteration variants, the probability calculated at a particular step of analysis is multiplied by a pre-selected weighting factor of the rule which is used at the given step of analysis to find the transliteration variant

One more optional feature of the method is that when the base with starting segments of words in the second language is complemented directly

during the comparison, if no variant of transliteration in the second language is found upon completing the analysis for the record in the first language being analyzed, the stepwise analysis of signals from the start to the end of the saved signal sequence can be repeated, and if this particular transliteration variant associated with the given step does not match any starting segment of some word in the second language, this transliteration variant is assigned a tentative occurrence frequency equal to a small preset value, at the same time transliteration variants having higher probability are saved.

Moreover, under the same situation, the number of transliteration variants that can be saved at each step of the analysis shall not exceed a preset number of the transliteration variants ending with the same "m – 1" characters of the second language and having higher probability.

Another optional feature of the method is that in the case that the transliteration variant having k chains is compared to a transliteration variant having q chains, where k and q are positive integers, the probabilities of both transliteration variants can be normalized by k and q, respectively, before the comparison, by extracting the k-th or q-th root, respectively, from the corresponding probability.

One more optional feature of the method is that beside the number of chains, the number of rules used in each of the compared variants can also be taken into account.

And at last, one more optional feature of the method is that a logarithmic measure of probability can be used as the probability of a transliteration variant.

The method can be implemented as first described above or with any number of the optional features described above. The steps of the method and all combinations of the optional features, can also be implemented corresponding systems, apparatus, and computer programs recorded on computer storage devices, each configured to perform the steps of the method

and any implemented optional features.

Detailed Description

This further description describes a detailed exemplary embodiment of the present invention method illustrated by an example of transliterating records with English (Latin) letters to records in Russian (Cyrillic letters). However, the example merely depicts the embodiment of the present invention method, and under no circumstance it shall be treated as a limitation for the method, the scope of which is determined only by the attached formula of invention.

The method actually begins with a preparatory stage, where any public or other databases are used to collect statistics of ways to write certain combinations of letters in one (first) language with letters of another (second) language. For the purpose of transliteration it is necessary to have two sets of data:

(1) *Transliteration rules*, i.e. a set of rules describing potential transliteration process. For example, «sh → ш». The rules need not be one-for-one (e.g. «s → с», «c → с»), and can even be overlapping (e.g. «s → с», «ch → ч», but «sch → щ»). One of the ways for taking overlaps into account involves assigning weighting coefficients to each rule. So, a transliteration rule represents a set that includes:

– a combination, i.e. a set of Latin letters denoting a piece of a record, that, once found in the record, may be replaced according to such a rule;

– contents, i.e. a chain of letters of the target language (in our examples – Russian), into which the combination is to be transformed; and

– weight, i.e. a rational number identifying significance of the rule among the other rules.

(2) *statistics of variants*, i.e. statistical information collected from various data bases and used to identify the best candidate – the one we consider the best

among others - of the given transliteration in the target language.

For the purpose of this description, "record" means the input of transliteration. It is a word of target language written with the alphabet of the first language. For example, we transliterate "mir"->"мир". "mir" is the (input) record, "мир" is its transliteration, the word of the target language (peace).

From the general point of view, everything seems simple. There is a set of rules (in the case under consideration – a set of presentations from the sequence of Latin letters to the sequence of Russian letters) that identifies the transliteration. Then, it is "only" required to apply each potential rule to each potential sequence to obtain a number of potential transliteration variants. However, this is not practical: for example, a record of the Russian word «zashtsheeshtshayoushtsheekhsya» provides 34 million transliteration variants.

As transliteration is generally ambiguous, a record could be transliterated in many inconsistent ways. For example, Russian is abundant in groups of obviously inconsistent rules:

- 'y' → 'и', 'y' → 'ий', 'y' → 'ый', 'y' → 'ы';
- 'ch' → 'ч', 'sch' → 'щ';
- ...

Thus, the record «schy» corresponding to word «щи» has at least $2 \times 4 = 8$ transliteration variants. Therefore, a certain method of selecting a preferable variant is required. Let us specify function μ for all such spelling variants, and compare $\mu(T_i)$ and $\mu(T_j)$, where T_i and T_j represent comparable spelling variants ($i \neq j$). One might assume that, if T is a dictionary word or the start of certain dictionary word, then $\mu(T) = 1$, otherwise, $\mu(T) = 0$. However, any dictionary contains a limited number of words (by which we mean combinations of letters) used in the given language. The method is designed even for mistakenly written words. Therefore, a different approach will be used.

Let us select a certain integer number **m**, and introduce the following definition:

$\mu(T) = \mu_T$, if length of **T** is less than or equal to **m**;

$\mu(T) = \mu(T_{[1..m]}) \times \mu(T_{[2..m+1]}) \times \ldots \times \mu(T_{[k..m+k-1]})$, where length of **T** is equal to $m + k - 1$, where the designation $T_{[a..b]}$ represents subsequence **T** in a word from position a to position b inclusive.

Thus, it is required to define all constants μ_T for complete definition of $\mu(T)$ for any **T**. Let us assume that μ_T (for any subsequence **T** being as long as $\leq m$) represents a number of subsequences **T** found as a subsequence of the target language words in the target language analyzed texts taken from a certain source, for example, from Internet web pages.

Quantitative values (μ_T) can always be transformed into a probability by dividing each such value by total sum. In this case, function μ is determined in all variants **T** of the target language, and expresses similarity degree of **T** to words from the analyzed source. It can be used as a probability measure of the target language variants.

An m-long piece of the target language word will be referred to as an *m-chain* (or just a chain). The term *m-chain* will also be used to refer to the start of a word that is less than m characters long, to be able to identify the probability of a variant having failed to reach a length of m characters. A suitable value for m depends on the language and can be determined heuristically, for example. On the one hand, m should be a small number because total number of chains to be saved in the memory grows like a power m. On the other hand, given too small m, the words with frequently occurring chains can prove preferable to the correct ones. For example, let the correct variant of a record transliteration be «abcdef» for a language with m = 3, though, if, for instance, μ[def] < μ[del], then variant «abcdel» will prove preferable.

For Russian, it works well to take m equal to 6. Let us note that when using text from the Internet, not every chain is used. To cut out rare words and obtain more consistent data, a minimum frequency or occurrence threshold is applied.

In comparing two spelling variants and picking the best one, one cannot simply compare probabilities (**μ**) if such variants differ in length, since the shorter variant has higher probability due to the lower number of multipliers in the product. A special function is used for calculating difference between such lengths and for multiplying probabilities of the shorter variant by mean probability used in creating variants to be compared. Thereby, a normalization of the variants are attained as if their lengths are equal.

For further description, let us assume that each record in the source (first) language is provided at the start and at the end with certain preset characters. The method does not require such additional characters, but they are fairly helpful for illustrating its embodiment. These characters can be either similar or different, either alphabetical or non-alphabetical. The next illustrative example uses character «^» at the start of the record and character «$» at the end of the word. Then, for example, 5-chain [^abcd] denotes sequence of letters at the start of the word, while 5-chain [^who$] denotes individual word «who».

Let us introduce another definition: end of **T** transliteration variant is represented by the last m – 1 letters in **T**, where m is the size of chains used for this language. Let us also point out that, if variant **T** is shorter than m letters, its end will be represented by variant **T** itself. For example, when 6-chains are used for Russian, then the end for variant **T** = «^метропо» will be represented by [тропо].

It should be noted that hereinafter records of the words in the first language are termed for the sake of simplicity as words, although in the strict sense they do not constitute words.

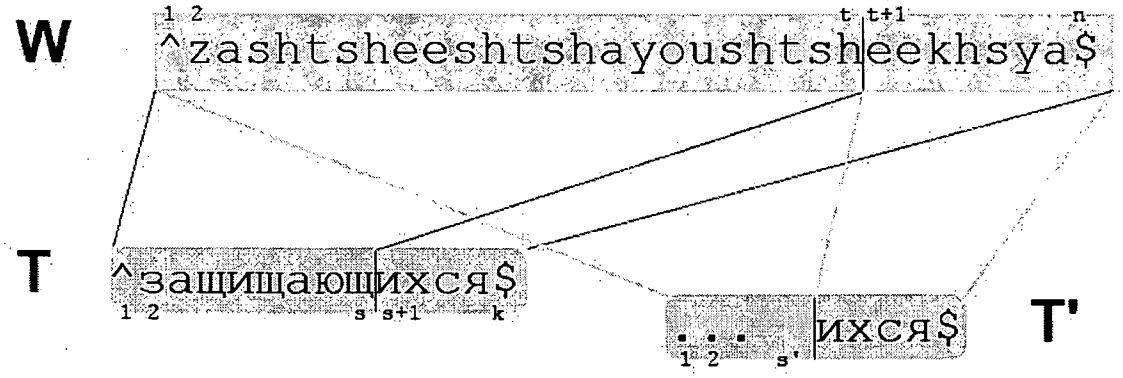
The following result – presented as a theorem – will be referred to later in this specification.

Theorem. Let us assume that there are statistics of a language with probability measure μ on the basis of m-chains. Let there be a record **W** as long as n, and its best (according to μ) transliteration variant **T** = **T**(**W**) as long as k.

For any positive integer t ($t \leq n$), let us consider $\mathbf{W}_{[1..t]}$ – the initial segment of record **W** until position t inclusive. $\mathbf{T}_{[1..s]}$ is the part of **T** corresponding to transliteration $\mathbf{W}_{[1..t]}$.

Then $\mathbf{T}_{[1..s]}$ is the better (according to μ) variant among all other variants with the same end of transliteration $\mathbf{W}_{[1..t]}$.

Example. For Russian language (**m** = 6) and record **W** = «^zashtsheeshtshayoushtsheekhsya$» the best variant **T** = «^защищающихся$». For t=24, $\mathbf{W}_{[1..24]}$ = «^zashtsheeshtshayoushtsh», and corresponding to it there is sub-variant (s=9) $\mathbf{T}_{[1..9]}$ = «^защищающ». Then $\mathbf{T}_{[1..9]}$ is the best variant among all other variants with the same end «ищающ» of transliteration $\mathbf{W}_{[1..24]}$.



Proof. Let us assume that for $\mathbf{W}_{[1..t]}$ there is another variant $\mathbf{T}'_{[1..s']}$ with the same end, and $\mu(\mathbf{T}'_{[1..s']}) > \mu(\mathbf{T}_{[1..s]})$. That is, . variant $\mathbf{T}'_{[1..s']}$ is better than $\mathbf{T}_{[1..s]}$.

Let us develop **T'** as concatenation $\mathbf{T}'_{[1..s']} + \mathbf{T}_{[s+1..k]}$. Then, **T'** will also

constitute a variant of transliteration **W** as a fusion of word **W** transliteration before position t and after it.

Then, considering that $\mathbf{T}_{[1..s]}$ and $\mathbf{T'}_{[1..s']}$ under hypothesis have the same ends, we obtain:

$$\mu(\mathbf{T}) = \mu(\mathbf{T}_{[1..s]}) \times \mu(\mathbf{T}_{[s-m+1..s+1]}) \times \mu(\mathbf{T}_{[s-m+2..s+2]}) \times \ldots \times \mu(\mathbf{T}_{[s..s+m]}) \times \mu(\mathbf{T}_{[s+1..k]}),$$

$$\mu(\mathbf{T'}) = \mu(\mathbf{T'}_{[1..s']}) \times \mu(\mathbf{T}_{[s-m+1..s+1]}) \times \mu(\mathbf{T}_{[s-m+2..s+2]}) \times \ldots \times \mu(\mathbf{T}_{[s..s+m]}) \times \mu(\mathbf{T}_{[s+1..k]})$$

Note that if s<m, s'<m or s+m>k, certain multipliers will disappear.

Since, by hypothesis, the first element of the product for $\mu(\mathbf{T'})$ is larger than the first element for $\mu(\mathbf{T})$, and the rest of them are equal, then $\mu(\mathbf{T'}) > \mu(\mathbf{T})$. We have a contradiction with the hypothesis that **T** is the best variant for **W**.

It should be noted that this result is true even if all the words are short (having length less than **m**).

Thus, the algorithm alone as embodied by the present method is fairly simple. As an option in the illustrative example, the record in the first language has signs «^» and «$» added to indicate the start and the end of this record. Then, the record will undergo stage-by-stage analysis from the first to the last letter (including the above added signs). In reviewing the position of each subsequent letter, all suitable rules will be applied starting from the current position, which results in a number of transliteration variants at the given stage of the analysis. Then, the rules will be applied to previously obtained variants, and eventually the variants of the entire word will be obtained.

At each stage, only variants with unique ends will be retained. If some variants have the same ends, the most probable variant will be retained. (In the event that **M** of the most probable resultant variants are required, then the largest **M** of the best variants with the same end shall be retained). We are entitled to this according to the main theorem inference saying that the best variants will arise out of the best sub-variants (among those having the same ends).

Let us point out that sub-variants having different ends are not compared with one another. Let us assume that variant «abc» is better than variant «aBC», but the application of the following rule may make variant «abcd» inferior to variant «aBCd». Therefore, if the ends are different, we retain all the variants in the hope that some of them can give the best estimate.

Retaining only the best variants at each stage will make our working set as small as required, and shall refrain us from iteration at each possible transliteration.

Each time, when we apply a certain rule to a certain variant, we update its probability according to the chains that we added, and multiply it by weight of the rule.

Example. Let **m** = 6 for Russian; and let the input record **W** = «^metropol$».

Assume that after transliteration «^metropo», we obtain variant {^метропо}. One of the rules that corresponds to the last letter «l» is rule **R**: «l» → «ль». Then, we add {ль}, thus obtaining «^метрополь», and have to recalculate its frequency multiplying it by **μ**[тропол] × **μ**[рополь] × weight (**R**).

Finally, we use the most frequent variant or variants that we obtained at the final stage, and remove signs «^» and «$».

Now, let us consider implementation of this algorithm in the present method.

As this method is designed for automatic transliteration, all the records to be transliterated are made suitable for computer processing. Namely, the records are coded in any acceptable code (e.g. ASCII, KOI-8, Unicode etc.) and transformed into appropriate signals to be transmitted through communication lines or recorded to a machine-readable medium. A sequence of such signals, with the help of which the first language record to be transliterated was coded

as a sequence of characters, will be received for its further processing in a duly programmed data processing unit. The latter can be a personal computer (PC), personal digital assistant (PDA), mobile telephone, server or any other device that can be provided with a program written according to a method stated in the present description. This received sequence of signals will optionally be supplemented at its start and end with signals corresponding to preset characters. As already noted above, the characters can be similar or different, alphabetic or non-alphabetic. In the latter case, for example, all characters of a record to be transliterated can be lower-case characters, while the start and end of such word can be supplemented with capital letters. The method will be performed on this data.

To do this, a first character will initially be selected, and its appropriate transliteration rule will be searched. Normally, there are rules that transliterate characters designating start and end of a record respectively into themselves. However, a rule can be introduced that will be applicable to only the record character, should it rank first. For example, ^i → ^й. The last characters will be treated similarly, e.g. yov$ → ев$. Then, the second character (i.e. the first letter of the record) will be chosen, and transliteration rules corresponding to it will be searched. Each found rule, i.e. corresponding character or a chain of corresponding characters (if a character of the source language can be transliterated by means of, say, two characters of the target language – e.g. «l» → «ль») will be used to supplement each transliteration rule found in the previous iteration. In the given case, a non-alphabetic character, introduced at the record start and being transliterated into itself, will be supplemented with characters of the second language from each transliteration rule found at this stage for the first character of the source language. In principle, a record in the first language can use characters selected from a group made up of, for example, the following: alphabetic characters of the first language; non-

alphabetic characters, each of which bears resemblance to one of alphabetic characters in the second language (for example, figure «3» can be used to record Russian letter «з» instead of Latin letter «z»); non-alphabetic characters, the combination of which bears resemblance to one of alphabetic characters in the second language (for example, a pair of characters «/\» (forward and backward slash characters) could be used to record Russian letter «л» instead of Latin letter «l»).

The next stage of the analysis involves picking up the second letter of the record being transliterated (i.e. the third character, should such record start and end be supplemented with non-alphabetic characters), and searching transliteration rules for such second letter. All thus found characters of the second language will supplement the chains previously made up of the second language characters, i.e. the chains of the record starting non-alphabetic character supplemented with the second language appropriate characters found for the first letter of the record in the source (first) language.

Let us also point out that the rules are applicable not only to individual characters of the record, but to groups of characters starting with the analyzed character. In this case, engendered variants fall into a stage preceding another stage that will involve consideration of a record character following those transformed by the rule.

These analysis stages are repeated until no more characters of the record being transliterated remain, i.e. in the present case, to a non-alphabetic character added to the end of such record. At each stage of analysis, each chain already made up at the previous stage of the second (target) language characters will be supplemented with characters found at the present stage. Thereafter, each of the resulting chains of target language characters are compared by reference to the appropriate base of initial segments with initial segments of the second language words. Such a comparison process involves picking-up, at

each stage, the variants having higher occurrence frequency as compared with other variants having the same ends out of many transliteration variants found in the course of the above comparison:

1. We parse the input record from the beginning to the end character-by-character.

2. At each stage, we find all rules matching the input record from the current character and on.

3. All chains already made-up at previous stage are supplemented with the contents of the found rules.

4. By referring to the base of initial segments of the second language words, we assign the probability of the variant. If no initial segment equal to the variant found, the variant is dropped.

5. The variants with same endings (same last m-1 characters) and on the same stage, are compared by their probability. The higher probability variant is saved, others dropped.

In principle, the base of the second language initial segments can be made up beforehand using a data collection approximating all words in the second language. However, this would require a very voluminous database. Therefore, it is more efficient to follow another way, and to form this base of initial segments of the second language words just in the course of the above comparison with initial segments of real words in the second language. Such a base of initial segments of the second language words will be compiled from real words in the second language, while initial segments of the second language words will be made up by mapping such chains with a single-character shift. Probability of each specific transliteration variant associated with the above stage and not exceeding m characters in length will represent occurrence frequency of an appropriate chain, while the probability of each specific transliteration variant associated with the above stage and exceeding m

characters in length will be found through multiplying occurrence frequencies of m-character long chains contained in the above-specified transliteration variant with overlapping due to shift of each next m-character long chains by one character from the start of the above specific transliteration variant. For example, for variant «абвгдеж», the previously found occurrence frequencies for chains «абвгд», «бвгде» and «вгдеж» will be multiplied. The resultant product will constitute the target probability for variant «абвгдеж». Each of the above chains having occurrence frequency in the second language lower then a preset limit can initially be removed from the chains' database; and if a certain chain is absent, the corresponding transliteration variant may be discarded.

In certain cases, it seems expedient, whilst calculating the probabilities of transliteration variants, to multiply the probability calculated at a specific stage of analysis by a previously selected weight coefficient assigned to the rule used at the above analysis stage to find transliteration variant. This helps to estimate variants based not only on its characters, but also on rules that generated the variant. It is useful in some cases, when exclusion of the above variants may result in failure to find transliteration for the entire record in the source language, since without assigning such a small probability, transliteration results obtained in the previous iterations may be discarded.

In the case when, on completion of the last stage of analysis, no transliteration variant in the second language for the analyzed record in the first language has been found, the above stage-by-stage analysis of signals will be repeated from the start to the end of the stored signal sequence, and where there is no coincidence of the above stage-specific transliteration variant with any initial segment of the word in the second language, such transliteration variant will be assigned fake occurrence frequency with preset low value. In this case, retained at each analysis stage will be no more than a preset number of transliteration variants having the same end of m – 1 characters of the second

language, and a frequency bring higher then the rest of the variants with the same ends of m – 1 characters. This would be helpful for the transliteration into such languages as Arabic, where the same letter appearance can vary depending on its position at the start, middle or end of a word. If the target language has no such peculiarities, the most probable variant will be chosen from transliteration variants with similar latest m – 1 characters of the target language.

In the case of comparing a transliteration variant consisting of k aforementioned chains and rules with a transliteration variant consisting of q aforementioned chains and rules, where k and q are positive integers, the probabilities of both transliteration variants will be normalized prior to comparison in terms of k or q, respectively, by extraction of k-th or q-th root, respectively, of the appropriate probability values.

For example, with initial record «^shkola$» at the third stage, we have variant V_1 = «^cx» consisting of a single chain and three rules: ^ → ^, s → c, h → x, and variant V_2 = «^ш» consisting of a single chain and two rules: ^ → ^, sh → ш. In this case, not the first root, i.e. not the probability values themselves (comparison frequency) shall be compared, but respectively, the fourth root of the occurrence frequency of variant V_1 and the third root of the occurrence frequency of variant V_2.

One may point out that it is expedient to use the logarithmic measure of frequency of occurrence of chains of the second language characters as a probability of a fairly small value due to huge numbers of words relied upon in creating the above base of rules and base of initial segments.

When the processing of the entire record in the source language to be transliterated is completed through multiple stages as described above, the final most probable result can be treated as a target transliteration in the target language. This is followed with generation of a signals sequence corresponding to the second language transliterated word encoding that, for example, can be

transmitted over communication lines or written in a respective memory.

The transliteration examples in accordance with the method described above are shown below. These examples represent the variants attributed to each transliteration stage and left (not discarded) for analysis at subsequent stages.

Example I.

Transliteration of record «olga».

End	Probability	Transliterated part	Applied rules (with weights)
1: ^lolga$			
[^]	1	"^"	^→^

At the first stage, initial signal was transliterated into itself.

2: ^ollga$			
[^o]	0,0198769	"^o"	^→^ o→o
3: ^ollga$			
[^ол]	0,000146119	"^ол"	^→^ o→o l→л
[^оль]	1,56053e–06	"^оль"	^→^ o→o l→ль w:0,05

At the third stage, the second letter of the record was transliterated in two ways giving rise to two variants. Since their ends differ, according to the theorem, no attempt at comparing and discarding has been made.

4: ^olgla$			
[^олг]	4,04846e–07	"^олг"	^→^ o→o l→л g→г
[^олж]	1,74308e–09	"^олж"	^→^ o→o l→л g→ж w:0,01

[^ольг]	1,33855e–06	"^ольг"	^→^ о→о l→ль w:0,05 g→г

At the fourth stage, next letter g, once also given two-fold transliteration, has engendered four variants. But, as chain [^ольж] has zero probability, its variant has been discarded.

5: ^olgal$

[^олга]	9,65577e–08	"^олга"	^→^ о→о l→л g→г a→a
[^олжа]	3,8423e–10	"^олжа"	^→^ о→о l→л g→ж w:0,01 a→a
[ольга]	1,11676e–06	"^ольга"	^→^ о→о l→ль w:0,05 g→г a→a

6: ^olga$

[льга$]	2,55749e–11	"^ольга$"	^→^ о→о l→ль w:0,05 g→г a→a $→$
[олга$]	3,4578e–08	"^олга$"	^→^ о→о l→л g→г a→a $→$

Result: **ольга**

Example II.

Transliteration of record «shashka».

This example illustrates application of rules assigned to higher-order stages than those, at which it was applied.

For example, the first character s and subsequent characters were analyzed at the second stage. One of the rules, sh→ш, affected the record second and third positions at once. Therefore, variant «^ш» engendered by them has been referred at once to the third stage.

1: ^|shashka$

End	Probability	Transliterated part	Applied rules (with weights)
[^]	1	^	^→^

2: ^s|hashka$

[^з]	8,10924e-05	^з	^→^ s→з w:0,01
[^с]	0,0309413	^с	^→^ s→с
[^съ]	5,14555e-05	^съ	^→^ s→съ
[^сь]	4,51423e-07	^сь	^→^ s→сь w:0,1
[^ш]	1,80712e-06	^ш	^→^ s→ш w:0,001

3: ^sh|lashka$

[^зх]	7,64281e-09	^зх	^→^ s→з w:0,01 h→х
[^сх]	9,61574e-05	^сх	^→^ s→с h→х
[^ш]	0,00180712	^ш	^→^ sh→ш
[^шх]	3,17676e-09	^шх	^→^ s→ш w:0,001 h→х
[^щ]	5,4019e-06	^щ	^→^

			sh→щ w:0,01

4: ^shalshka$

[^зха]	3,72714e-10	^зха	^→^ s→з w:0,01 h→x a→a
[^сха]	9,58236e-08	^сха	^→^ s→c h→x a→a
[^ша]	0,000238168	^ша	^→^ sh→ш a→a
[^шха]	1,97438e-11	^шха	^→^ s→ш w:0,001 h→x a→a
[^шя]	2,59001e-09	^шя	^→^ sh→ш a→я w:0,001
[^ща]	1,10777e-07	^ща	^→^ sh→щ w:0,01 a→a
[^щя]	2,93584e-11	^щя	^→^ sh→щ w:0,01 a→я w:0,001

5: ^shaslhka$

[^шас]	4,24794e-06	^шас	^→^ sh→ш a→a s→c
[^шаш]	6,59954e-09	^шаш	^→^ sh→ш a→a s→ш w:0,001
[^шяс]	2,01154e-11	^шяс	^→^ sh→ш a→я w:0,001 s→c
[^шяш]	6,9434e-13	^шяш	^→^

			sh→ш a→я w:0,001 s→ш w:0,001
[^щаз]	2,41509e-11	^щаз	^→^ sh→щ w:0,01 a→a s→з w:0,01
[^щас]	4,47293e-08	^щас	^→^ sh→щ w:0,01 a→a s→c
[^щаш]	1,63144e-12	^щаш	^→^ sh→щ w:0,01 a→a s→ш w:0,001
[^щяз]	1,94088e-15	^щяз	^→^ sh→щ w:0,01 a→я w:0,001 s→з w:0,01
[^щяс]	7,63459e-13	^щяс	^→^ sh→щ w:0,01 a→я w:0,001 s→c
[^щяш]	1,06198e-15	^щяш	^→^ sh→щ w:0,01 a→я w:0,001 s→ш w:0,001

6: ^shashlka$

[^шаш]	6,59954e-06	^шаш	^→^ sh→ш a→a sh→ш
[^шащ]	2,84768e-09	^шащ	^→^ sh→ш a→a sh→щ w:0,01
[^шяш]	6,9434e-10	^шяш	^→^ sh→ш a→я w:0,001 sh→ш

[^шящ]	4,95191e-12	^шящ	^→^ sh→ш a→я w:0,001 sh→щ w:0,01
[^щаш]	1,63144e-09	^щаш	^→^ sh→щ w:0,01 a→a sh→ш
[^щащ]	8,60848e-11	^щащ	^→^ sh→щ w:0,01 a→a sh→щ w:0,01
[^щяш]	1,06198e-12	^щяш	^→^ sh→щ w:0,01 a→я w:0,001 sh→ш
[^щящ]	8,05361e-14	^щящ	^→^ sh→щ w:0,01 a→я w:0,001 sh→щ w:0,01

7: ^shashkla$

[^шашк]	2,52467e-06	^шашк	^→^ sh→ш a→a sh→ш k→к

8: ^shashkal$

[шашка]	5,38716e-07	^шашка	^→^ sh→ш a→a sh→ш k→к a→a

9: ^shashka$|

[ашка$]	1,03159e-13	^шашка$	^→^ sh→ш a→a sh→ш k→к

			a→a $→$

Result: **шашка**

Example III.

Transliteration of record «podzol».

This example illustrates repeated transliteration process due to the absence of a result following the first passage. Indeed, the probability of chain [одзол$] is zero, and expected result «подзол» was discarded at the last stage. The other variants were also discarded at various stages for the same reason.

The second time, the variants that failed to coincide with any initial segments of words in the second language, were assigned low probability 1e-10. Since no variants have been discarded in the transliteration process, their total number is large, but the result was achieved.

1: ^|podzol$

End	Probability	Transliterated part	Applied rules (with weights)
[^]	1	^	^→^

2: ^p|odzol$

[^п]	0,0356523	^п	^→^ p→п
[^p]	1,40238e-07	^p	^→^ p→p(1e-05)

3: ^po|dzol$

[^по]	0,014841	^по	^→^ p→п o→o
[^po]	1,82471e-08	^po	^→^ p→p(1e-05) o→o

4: ^pod|zol$

[^под]	0,00224693	^под	^→^

			p→п o→o d→д
[^подъ]	5,03591e-06	^подъ	^→^ p→п o→o d→дъ(0,2)
[^подь]	8,71406e-08	^подь	^→^ p→п o→o d→дь(0,2)
[^род]	2,09531e-09	^род	^→^ p→p(1e-05) o→o d→д
[^родъ]	4,23415e-14	^родъ	^→^ p→p(1e-05) o→o d→дъ(0,2)
[^родь]	2e-16	^родь	^→^ p→p(1e-05) o→o d→дь(0,2)

5: ^podzlol$

[^подж]	1,52488e-07	^подж	^→^ p→п o→o d→д z→ж(0,03)
[^подз]	1,50648e-05	^подз	^→^ p→п o→o d→д z→з
[^подц]	2,20269e-08	^подц	^→^ p→п o→o d→д z→ц(0,05)
[^родж]	5,94019e-13	^родж	^→^

			p→p(1e-05) o→o d→д z→ж(0,03)
[^родз]	9,18225e-13	^родз	^→^ p→p(1e-05) o→o d→д z→з
[^родц]	5e-17	^родц	^→^ p→p(1e-05) o→o d→д z→ц(0,05)
[одъзь]	4e-22	^подъзь	^→^ p→п o→o d→дъ(0,2) z→зь(0,2)
[одьзь]	4e-22	^подьзь	^→^ p→п o→o d→дь(0,2) z→зь(0,2)
[подзь]	2e-11	^подзь	^→^ p→п o→o d→д z→зь(0,2)
[подъж]	6e-13	^подъж	^→^ p→п o→o d→дъ(0,2) z→ж(0,03)
[подъз]	2e-11	^подъз	^→^ p→п o→o d→дъ(0,2) z→з
[подъц]	1e-12	^подъц	^→^ p→п

			o→o d→дъ(0,2) z→ц(0,05)
[подьж]	6e-13	^подьж	^→^ p→п o→o d→дь(0,2) z→ж(0,03)
[подьз]	2e-11	^подьз	^→^ p→п o→o d→дь(0,2) z→з
[подьц]	1e-12	^подьц	^→^ p→п o→o d→дь(0,2) z→ц(0,05)
[родзь]	2e-16	^родзь	^→^ p→p(1e-05) o→o d→д z→зь(0,2)
[родъж]	6e-18	^родъж	^→^ p→p(1e-05) o→o d→дъ(0,2) z→ж(0,03)
[родъз]	2e-16	^родъз	^→^ p→p(1e-05) o→o d→дъ(0,2) z→з
[родъц]	1e-17	^родъц	^→^ p→p(1e-05) o→o d→дъ(0;2) z→ц(0,05)
[родьж]	6e-18	^родьж	^→^ p→p(1e-05) o→o

			d→дь(0,2) z→ж(0,03)
[родьз]	2e-16	^родьз	^→^ p→p(1e-05) o→o d→дь(0,2) z→з
[родьц]	1e-17	^родьц	^→^ p→p(1e-05) o→o d→дь(0,2) z→ц(0,05)

6: ^podzoll$´

[дъзьо]	4e-32	^подъзьо	^→^ p→п o→o d→дъ(0,2) z→зь(0,2) o→o
[дьзьо]	4e-32	^подьзьо	^→^ p→п o→o d→дь(0,2) z→зь(0,2) o→o
[одзьо]	2e-21	^подзьо	^→^ p→п o→o d→д z→зь(0,2) o→o
[одъжо]	6e-23	^подъжо	^→^ p→п o→o d→дъ(0,2) z→ж(0,03) o→o
[одъзо]	2e-21	^подъзо	^→^ p→п o→o

			d→дъ(0,2) z→з o→o
[одъцо]	1e-22	^подъцо	^→^ p→п o→o d→дъ(0,2) z→ц(0,05) o→o
[одьжо]	6e-23	^подьжо	^→^ p→п o→o d→дь(0,2) z→ж(0,03) o→o
[одьзо]	2e-21	^подьзо	^→^ p→п o→o d→дь(0,2) z→з o→o
[одьцо]	1e-22	^подьцо	^→^ p→п o→o d→дь(0,2) z→ц(0,05) o→o
[поджо]	2,73535e-08	^поджо	^→^ p→п o→o d→д z→ж(0,03) o→o
[подзо]	7,79656e-07	^подзо	^→^ p→п o→o d→д z→з o→o
[подцо]	5e-12	^подцо	^→^ p→п

			o→o d→д z→ц(0,05) o→o
[роджо]	3e-17	^роджо	^→^ p→p(1e-05) o→o d→д z→ж(0,03) o→o
[родзо]	1e-15	^родзо	^→^ p→p(1e-05) o→o d→д z→з o→o
[родцо]	5e-17	^родцо	^→^ p→p(1e-05) o→o d→д z→ц(0,05) o→o

7: ^podzoll$

[джоль]	1,36767e-29	^поджоль	^→^ p→п o→o d→д z→ж(0,03) o→o l→ль(0,05)
[дзоль]	3,11498e-25	^подзоль	^→^ p→п o→o d→д z→з o→o l→ль(0,05)
[дзьол]	2e-31	^подзьол	^→^ p→п o→o

			d→д z→зь(0,2) o→o l→л
[дцоль]	2,5e-33	^подцоль	^→^ p→п o→o d→д z→ц(0,05) o→o l→ль(0,05)
[дъжол]	6e-33	^подъжол	^→^ p→п o→o d→дъ(0,2) z→ж(0,03) o→o l→л
[дъзол]	2e-31	^подъзол	^→^ p→п o→o d→дъ(0,2) z→з o→o l→л
[дъцол]	1e-32	^подъцол	^→^ p→п o→o d→дъ(0,2) z→ц(0,05) o→o l→л
[дьжол]	6e-33	^подьжол	^→^ p→п o→o d→дь(0,2) z→ж(0,03) o→o l→л
[дьзол]	2e-31	^подьзол	^→^ p→п

			o→o d→дь(0,2) z→з o→o l→л
[дьцол]	1e-32	^подьцол	^→^ p→п o→o d→дь(0,2) z→ц(0,05) o→o l→л
[зьоль]	1e-42	^подзьоль	^→^ p→п o→o d→д z→зь(0,2) o→o l→ль(0,05)
[оджол]	2,73535e-18	^поджол	^→^ p→п o→o d→д z→ж(0,03) o→o l→л
[одзол]	6,22996e-14	^подзол	^→^ p→п o→o d→д z→з o→o l→л
[одцол]	5e-22	^подцол	^→^ p→п o→o d→д z→ц(0,05) o→o l→л
[ъжоль]	3e-44	^подъжоль	^→^

			p→п o→o d→дъ(0,2) z→ж(0,03) o→o l→ль(0,05)
[ъзоль]	1e-42	^подъзоль	^→^ p→п o→o d→дъ(0,2) z→з o→o l→ль(0,05)
[ъзьол]	4e-42	^подъзьол	^→^ p→п o→o d→дъ(0,2) z→зь(0,2) o→o l→л
[ъцоль]	5e-44	^подъцоль	^→^ p→п o→o d→дъ(0,2) z→ц(0,05) o→o l→ль(0,05)
[ьжоль]	3e-44	^подьжоль	^→^ p→п o→o d→дь(0,2) z→ж(0,03) o→o l→ль(0,05)
[ьзоль]	1e-42	^подьзоль	^→^ p→п o→o d→дь(0,2) z→з o→o l→ль(0,05)

[ьзьол]	4e-42	^подьзьол	^→^ p→п o→o d→дь(0,2) z→зь(0,2) o→o l→л
[ьцоль]	5e-44	^подьцоль	^→^ p→п o→o d→дь(0,2) z→ц(0,05) o→o l→ль(0,05)

8: ^podzol$|

[джол$]	2,73535e-28	^поджол$	^→^ p→п o→o d→д z→ж(0,03) o→o l→л $→$
[дзол$]	6,22996e-24	^подзол$	^→^ p→п o→o d→д z→з o→o l→л $→$
[дцол$]	5e-32	^подцол$	^→^ p→п o→o d→д z→ц(0,05) o→o l→л $→$
[жоль$]	1,36767e-39	^поджоль$	^→^ p→п

			o→o d→д z→ж(0,03) o→o l→ль(0,05) $→$
[золь$]	3,11498e-35	^подзоль$	^→^ p→п o→o d→д z→з o→o l→ль(0,05) $→$
[зьол$]	2e-41	^подзьол$	^→^ p→п o→o d→д z→зь(0,2) o→o l→л $→$
[цоль$]	2,5e-43	^подцоль$	^→^ p→п o→o d→д z→ц(0,05) o→o l→ль(0,05) $→$
[ъжол$]	6e-43	^подъжол$	^→^ p→п o→o d→дъ(0,2) z→ж(0,03) o→o l→л $→$
[ъзол$]	2e-41	^подъзол$	^→^ p→п o→o

			d→дъ(0,2) z→з o→o l→л $→$
[ъцол$]	1e-42	^подъцол$	^→^ p→п o→o d→дъ(0,2) z→ц(0,05) o→o l→л $→$
[ьжол$]	6e-43	^подьжол$	^→^ p→п o→o d→дь(0,2) z→ж(0,03) o→o l→л $→$
[ьзол$]	2e-41	^подьзол$	^→^ p→п o→o d→дь(0,2) z→з o→o l→л $→$
[ьоль$]	1e-52	^подзьоль$	^→^ p→п o→o d→д z→зь(0,2) o→o l→ль(0,05) $→$
[ьцол$]	1e-42	^подьцол$	^→^ p→п o→o d-→дь(0,2)

			z→ц(0,05) o→o l→л $→$

Result: **подзол**

The above examples indicate not only operability of the proposed transliteration method, but also the potential for improving the accuracy and clarity of transliteration in automatic systems, i.e. systems operating without human intervention.

Although the claimed methods are described above by means of exemplary embodiments, these exemplary embodiments can be modified without departure from the spirit and scope of the invention, imposing no limitations on the scope of claims.

Claims

1. A method for automatic transliteration of a record in a first language to a word in a second language, which involves the following:

- receiving a sequence of signals, each of which encodes the corresponding character in the record in the first language to be transliterated, and saving the said received signal sequence to memory;

- during stepwise analysis of the signals from the start to the end of the saved signal sequence, finding all transliteration rules for the character being analyzed or character group starting with the character being analyzed, in the record in the first language to be transliterated, by looking up in the appropriate rule base created in advance;

- complementing all the transliteration variants obtained at the previous step with characters of the second language from each transliteration rule found at the given step, thus obtaining at the given step composite variants of transliteration associated with this step or with one of the subsequent steps;

- at each step, comparing each of the aforementioned composite transliteration variants obtained at the given step against starting segments of words in the second language, by looking up in the created base of starting segments of words in the second language, and if a particular transliteration variant associated with the given step matches a starting segment of a word in the second language, retaining this transliteration variant for analysis in subsequent steps;

- on completion of the last step of the analysis, accepting at least one transliteration variant in the second language associated with the last step as the transliteration result of the said record in the first language;

- creating a sequence of signals, each of which encodes the corresponding character in the resulting transliterated word of the second

language.

2. A method of claim 1, further comprising complementation of the said received signal sequence at its start and end with signals corresponding to preset characters, and the first step of analysis is started from the first of the said preset characters.

3. A method of claim 2, wherein the signals that complement the received signal sequence correspond to preset alphabetic characters.

4. A method of claim 2, wherein the signals that complement the received signal sequence correspond to preset non-alphabetic characters.

5. A method of claim 3 or 4, wherein the signals that complement the received signal sequence correspond to different preset characters.

6. A method of claim 3 or 4, wherein the signals that complement the received signal sequence correspond to the same preset character.

7. A method of claim 1, wherein the characters used for the record in the first language are selected from the group comprising: alphabetic characters of the first language; non-alphabetic characters, each resembling some alphabetic character of the second language; non-alphabetic characters that in combination resemble some alphabetic character of the second language.

8. A method of claim 1, wherein the said base of starting segments of words in the second language is created in advance.

9. A method of claim 1, wherein the said base of starting segments of words in the second language is created directly during the said comparison, from chains no longer than "m" characters, which chains are obtained from actual words of the second language, and the said starting segments of words in the second language are created by mapping the said chains with a shift by one character, wherein $m > 1$ is an integer pre-selected for the second language; during this process, the occurrence frequency in the second language is determined for each of the said chains, and the chains with occurrence

frequency below a preset limit are removed.

10. A method of claim 9, further comprising determination of the probability of each particular transliteration variant associated with the given step and no more than "m" characters long, wherein said probability is determined by finding the occurrence frequency of the corresponding chain, while for each particular variant that is longer than "m" characters and is associated with the given step, its probability is determined by multiplying the occurrence frequencies of those involved said chains "m" characters long which belong to the given particular transliteration variant with overlapping upon shifting each subsequent chain "m" characters long by one character from the beginning of the given particular transliteration variant; if any of the chains is missing, the variant is discarded.

11. A method of claim 10, wherein, during determination of the said probabilities of transliteration variants, the probability calculated at a particular step of said analysis is multiplied by a pre-selected weighting factor of the rule which is used at the given step of analysis to find the transliteration variant.

12. A method of claims 10 or 11, wherein if no variant of transliteration in the second language is found upon completing the analysis for the record in the first language being analyzed, the said stepwise analysis of signals is repeated from the beginning to the end of the said saved signal sequence, and if this particular transliteration variant associated with the given step does not match any starting segment of some word in the second language, this transliteration variant is assigned a tentative occurrence frequency equal to a small preset value.

13. A method of claims 10 or 11, wherein the number of transliteration variants saved at each step of the analysis does not exceed a preset number, and wherein the said variants end in the same "m – 1" characters of the second language and have higher probabilities.

14. A method of claim 13, wherein in the case that the transliteration variant comprising "k" said chains is compared to a transliteration variant comprising "q" said chains, where "k" and "q" are positive integers, the probabilities of both transliteration variants are normalized by "k" and "q", respectively, before the comparison, by extracting the k-th or q-th root, respectively, from the corresponding probability.

15. A method of claim 14, wherein, beside the number of chains, the number of rules used in each of the compared variants is taken into account by adding the number of rules used therein to "k" and "q", respectively.

16. A method of claim 1, wherein the accepted transliteration result after the last step is the variant with the highest probability.

17. A method of any of claims 10-16, wherein the logarithmic measure of probability is used as the said probability of a transliteration variant.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/RU 2010/000013

A. CLASSIFICATION OF SUBJECT MATTER ***G06F 17/28*** *(2006.01)*

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 7/00, 7/02, 7/04, 7/06, 17/00, 17/20, 17/21, 17/27, 17/28, 17/30, 19/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
DWPI, Esp@cenet, K-PION, PAJ, RUPTO, USPTO, БД ФИПС

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5640587 A (OBJECT TECHNOLOGY LICENSING CORP.) 17.06.1997, столбец 4, строки 1-12, столбец 8, строки 1-9, столбец 9, строки 44-46, столбец 10, строки 1-11, фиг. 6	1-6, 8, 16
Y	US 2009/0012775 A1 (SHERIKAT LINK LETATWEER ELBARMAGUEYAT S.A.E.) 08.01.2009, абзацы [0012]-[0013], [0019], [0023]-[0024], [0053]-[0054], фиг. 4	1-6, 8, 16
A	US 7099876 B1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 29.08.2006	1-17
A	US 2006/0112091 A1 (HARBINGER ASSOCIATES, LLC) 25.05.2006	1-17
A	US 2008/0270111 A1 (RAM PRAKASH HANUMANTHAPPA) 30.10.2008	1-17

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is nit considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
08 September 2010 (08.09.2010)	23 September 2010 (23.09.2010)
Name and mailing address of the ISA/RU FIPS Russia, 123995, Moscow, G-59, GSP-5, Berezhkovskaya nab., 30-1 Facsimile No. 243-3337	Authorized officer S. Ilyasov Telephone No. (499) 240-2591

Form PCT/ISA/210 (second sheet) (July 2009)