Language Identification for Texts Written in Transliteration

Andrey Chepovskiy, Sergey Gusev, Margarita Kurbatova

Higher School of Economics,
Data Analysis and Artificial Intelligence Department,
Pokrovskiy boulevard 11, 109028 Moscow, Russia
achepovskiy@hse.ru, unk379@mail.ru, rityastik@gmail.com

Abstract. The problem of identification of natural languages for the texts written in transliteration is considered. We consider a method of identification of five Slavic languages for texts written with use of a Latin transliteration. We use two ways of creation models for such texts and compare results of those application.

Keywords: statistical text model; natural language identification; transliteration.

1 Introduction

In various type of information systems intended for automatic processing of large amounts of texts in natural languages various data recognition problems are actual. The requirement of automating textual data processing brings specific importance to the language identification problem of a text or a part of a text.

At present time the sufficiently accurate language recognition methods for long texts consisting of tens of sentences are known [1]. Models based on frequencies of letter combinations are widely used to identify language of a text [2, 3]. It was noted in [3] that it is possible to use rank methods for language identification of a text, but they are not suitable for short texts. Also in [3] it is concluded that the language identification problem for short texts segments is still actual, and higher accuracy is achieved at the expense of larger model size and slower processing.

In [4] a language identification method for texts in natural language was studied. This method was applied to texts written in native alphabets for corresponding languages and good results were achieved. In current work we consider using the same method for language identification of a text written in Latin transliteration.

We use a set of five Slavic languages: Russian, Ukrainian, Byelorussian, Bulgarian and Macedonian. All of these languages use Cyrillic alphabet as native, and transliteration must be used to write in Latin.

The main problems related to language identification process are:

- A wide variety of conflicting transliteration tables;

- A frequent use of transliteration rules which do not meet any of standard transliteration tables.
- Amount of texts in transliteration is not enough for construction of training sets that are used in language identification algorithms.

To solve the first problem several transliteration tables should be used for each of the languages. Training text sets were created with an automatic text generation process from Cyrillic texts by using transliteration tables.

2 Statistical model for a natural language text

The text language identification problem is a pattern recognition problem, and its solution can be based on a probabilistic model. A Bayesian classifier can be applied to a string of characters assuming that we know statistical characteristics of characters for texts in specific language or texts belonging to a given class.

Let's consider a string s which consists of N characters c_n $(n = 1,\ldots, N)$ that belong to alphabet Σ. Further we will use the following notation: $s = c_1c_2\ldots c_N$ - a specific value of the string, $s_i = c_i$ – a value of a character which is at i-th position in the string. For solving the problem this string must be assigned to one of the classes $Y_l (l = 1, \ldots, K)$, where Y_l denotes one of K languages.

We assume that every class defines some kind of probability distribution on the set of all possible strings. In that case it is possible to apply a statistical criterion of maximum likelihood to determine a class which contains the string being classified.

The probability of a fact that string s will appear in some language equals to product of probabilities that each character of this string will appear in this language provided that all preceding characters will appear in this language too:

$$\begin{aligned} &P(s = c_1 \ldots c_N) = P(s_N = c_N \mid s_1 = c_1, \ldots, s_{N-1} = c_{N-1}) * \\ &P(s_{N-1} = c_{N-1} \mid s_1 = c_1, \ldots, s_{N-2} = c_{N-2}) * \ldots * P(s_1 = c_1) \end{aligned} \quad (1)$$

Let's assume that the probability distribution for a character at i-th position depends on probability distribution of not more than k preceding characters. In this case equation (1) can be written as following:

$$P(s_i = c_i \mid s_1 = c_1, \ldots, s_{i-1} = c_{i-1}) = P(s_i = c_i \mid s_{i-k} = c_{i-k}, \ldots, s_{i-1} = c_{i-1}) \quad (2)$$

An estimation of the conditional probabilities is performed on the training set. For this purpose, the frequencies of all substrings of lengths less than $k+2$ are calculated, and the estimated value of conditional probability for the next character is a ratio of the frequencies of the corresponding substrings:

$$P(Y_l, s_i = c_i \mid s_{i-m} = c_{i-m}, \ldots, s_{i-1} = c_{i-1}) = \frac{f(c_{i-m} \ldots c_i)}{f(c_{i-m} \ldots c_{i-1})}, \forall m \le k, \quad (3)$$

f(X) – the frequency of substring ***X*** in the training set.

The estimated value of probability of string ***s*** appearance in class Y_l is defined as follows:

$$P(Y_l, s) = P(Y_l, s_N = c_N \mid s_{N-k} = c_{N-k}, ..., s_{N-1} = c_{N-1}) * \\ P(Y_l, s_{N-1} = c_N \mid s_{N-k-1} = c_{n-k-1}, ..., s_{N-2} = c_{N-2}) * ... * P(Y_l, s_1 = c_1) \quad (4)$$

The classified string is assigned to the class with the highest probability estimate.

3 Algorithm implementation

First, each natural language text is converted to a set of words consisting of lower cased characters belonging to the native alphabet of the language. It forms a frequency dictionary of substrings having lengths in the range ***[1, k+1]*** taken into account the number of word occurrences in the text. This process is executed during the construction of a string model for a given language. This construction is based on training set of texts. The model is represented as a finite state machine with states marked with preceding character sequences and transitions marked with the next character and corresponding conditional probability.

A single space is appended to the end of each word of the text, then the word is passed as input to the finite state machine. The initial state of the machine corresponds to character sequence consisting of ***k*** spaces. According the formulas (1) – (4) a probability of transition to the next state from the current one by each of the characters is being calculated. A probability of appearance of the given word is a product of probabilities of all transitions that occurred during the machine operation. The probability of a text is a product of probabilities of all its words.

For the language identification a probability of the text appearance for models of every natural language is estimated. The language of the model with the highest probability is assigned to the text.

4 Quality of the text language identification

For the language identification of a text fragment a numeric estimate of its correspondence to a natural language text model can be calculated. Let the text fragment s consist of ***N*** characters. A probability of its appearance in the text written in ***l***-th language can be estimated with the formula (4). Then an estimation of this text fragment correspondence to the ***l***-th language will be calculated as:

$$E_l(s) = \frac{\ln(P(Y_l, s))}{N} + const, \quad (5)$$

P(Yl, s) – the probability of the string ***s*** appearance in the language Y_l;

N – the number of characters in the string ***s***;
const – a normalizing constant.
The expected value of this estimation doesn't depend on the length of text fragment. We choose the language with a maximum value of the estimation $E_l(s)$.

5 Initial data

Transliteration tables for all of five languages were constructed. Several different tables were used for each language – from 4 to 10, depending on language.

Cyrillic text sets consisting of at least 500 thousands characters were made for each of the five languages. These sets were automatically transliterated into Latin alphabet with each of the transliteration tables. Some tables contain ambiguous translation rules (i.e. there are several versions of Latin character combinations for one Cyrillic character). When several different rules were possible then only one of the rules was chosen randomly. These texts subsequently were used as training texts for creation of text models.

Test sets consisting of at least 50 thousands characters were constructed for each language. Test sets are real texts written in transliteration taken from various sources. We will call texts written in transliteration as transliterated texts, and texts written in native alphabets of corresponding languages as native texts.

To compare language identification quality for transliterated texts with language identification quality for native texts the following training and text sets of the same size were made:

- Cyrillic sets for the five languages being considered;
- Sets for 31 languages which use Latin alphabet as native.

6 Experimental results

We evaluate the quality of our language identification method by calculating precision and recall for individual languages. When we identify the language of a text sample of known language we can determine whether it was identified correctly or not. For a set of test samples we know the number of correctly identified samples as well as the number of identification errors for each language. For a given language precision can be calculated as a ratio of the number of correctly identified text samples in this language to the overall number of samples identified to this language. Recall is a ratio of correctly identified text samples in this language to the number of all samples in this language in the test set. We use F-measure to combine precision and recall to a single value, which is defined as harmonic mean of precision and recall.

To evaluate the quality of the language identification method we created a set of language models. Each model was trained on a text from the training set and marked with the language of that text. Then several sets of text samples were created from the texts belonging to the test set. Each set included 1000 text samples for each language

of the same length. The samples were generated as text fragments starting from a randomly selected position inside text with the restriction that the position must be a beginning of a word. Every text sample was evaluated with every model and the language of the model with the highest estimate was chosen as the identified language of the sample. Then the values of F-measure were calculated for every language.

The set of models included 5 models for transliterated texts of Slavic languages as well as 31 models for native texts with languages which use Latin as their native alphabet. The test set contained 36 texts for the same languages.

We considered two methods of language identification for transliterated texts. The first method uses several text models for each language. Each model corresponds to a particular transliteration table for the language. The model is trained on a text which was generated using that transliteration table. So the set of models contains several different models for each language. If a text sample receives the highest estimate for any of the models for a particular language then the sample is identified as belonging to that language. The second method uses exactly one text model for each language. The model is trained on a text which is a concatenation of all texts generated for the language by using all its transliteration tables.

Figures 1 and 2 show dependence of the language identification F-measure for transliterated texts on the text sample length. Using several models for each language instead of one model does not lead to significant improvement of the text language identification.

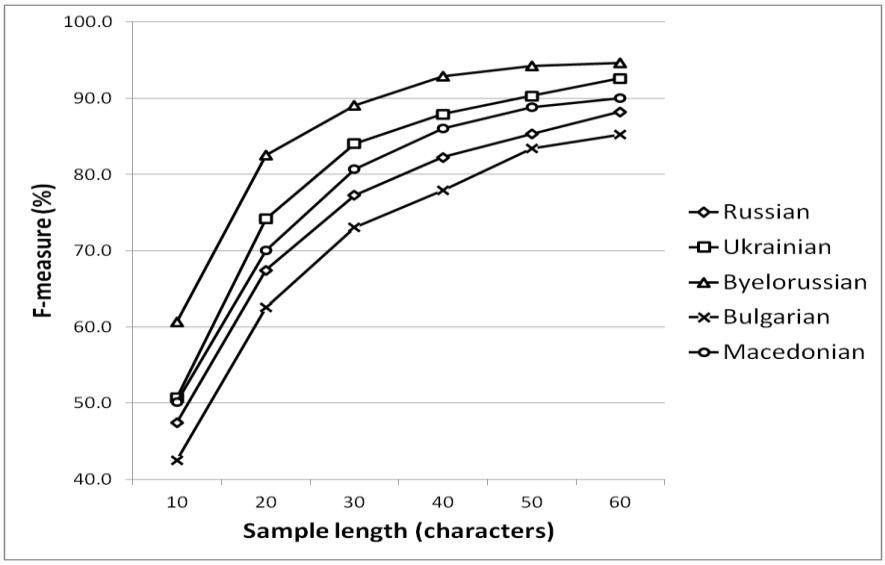

Fig. 1. Dependence of the F-measure on text sample length when using several models for one language.

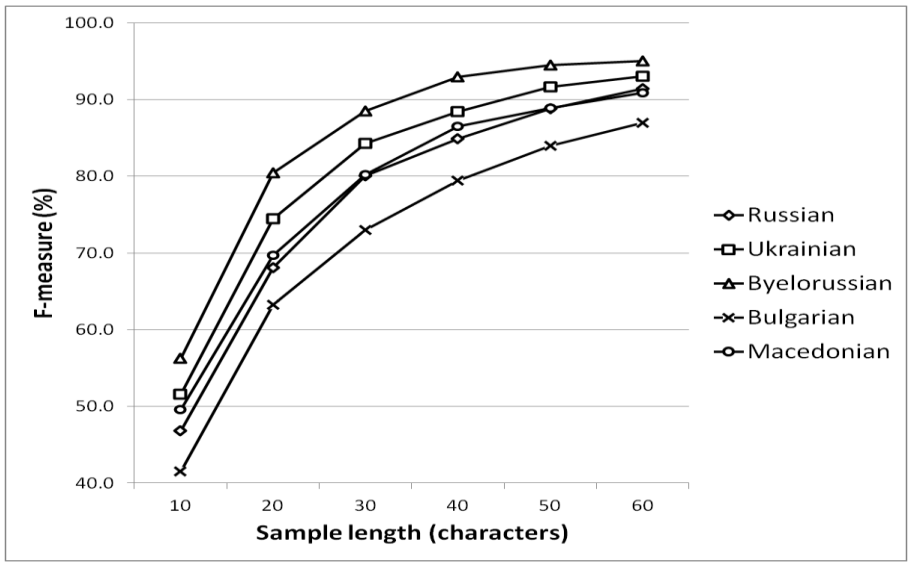

Fig. 2. Dependence of the F-measure on text sample length when using one model for one language.

For comparison figure 3 shows the language identification quality for native texts. Results were obtained by training and testing of five language models for the Cyrillic sets for the same five Slavic languages.

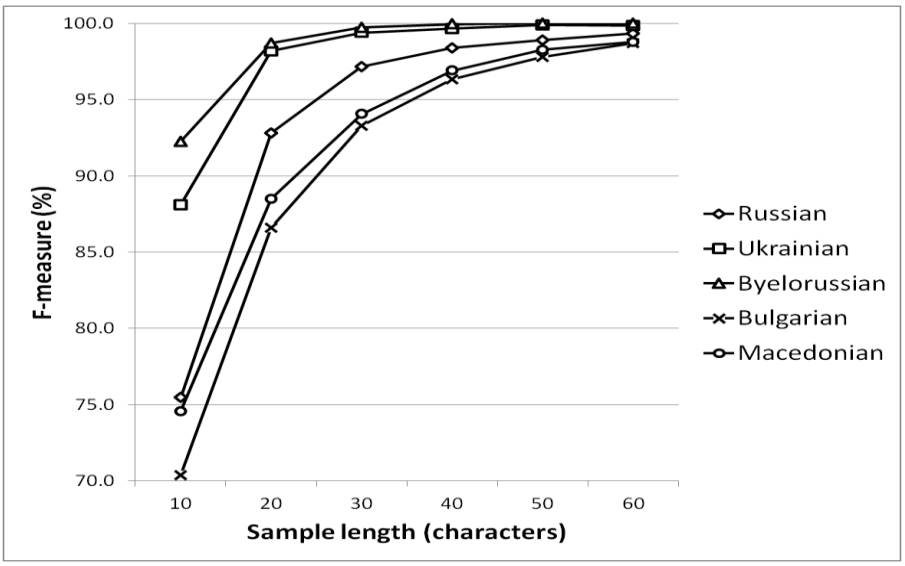

Fig. 3. Dependence of the F-measure on text sample length for language identification of native texts.

From figure 3 one can see that language identification quality for transliterated texts is significantly lower than for native texts. The most substantial part of accuracy loss occurs when transliterated text incorrectly classifies to model representing transliterated text of another language. There are much fewer errors when transliterated text classifies to a language that uses Latin alphabet.

To illustrate this fact, we can measure the quality of separation of transliterated texts from native texts written in Latin. It is much higher than language identification quality for transliterated text. Figure 4 shows F-measure values for separation of transliterated texts for various text sample lengths.

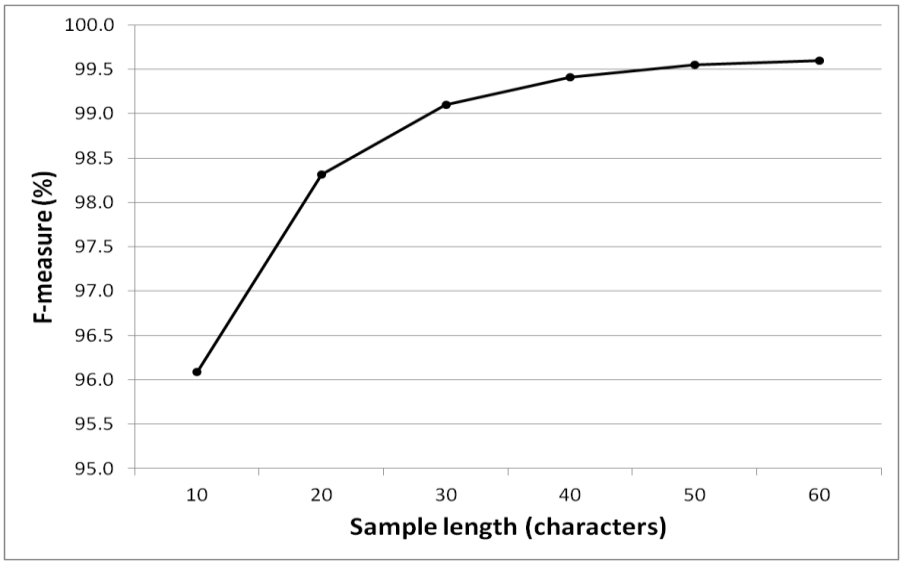

Fig. 4. F-measure values for separation of transliterated texts for various text sample lengths.

7 Conclusion

The text model considered in this article allows to successfully separate texts written in languages with Latin-based alphabets from texts written in Latin transliteration for languages with Cyrillic alphabets. The F-measure of such separation reaches 98% for short texts consisting of only 20 characters.

Identification accuracy of languages using Cyrillic alphabet for transliterated texts reaches more than 80% for texts consisting of 40 characters.

It was observed that using several different text models for a single language does not give a significant advantage over using a single model for a language.

References

1. McNamee, B.P.: Language identification: a solved problem suitable for undergraduate instructionl Journal of Computing Sciences in Colleges, 20(3), P. 94–101 (2005)
2. Cavnar, W. B., Trenkle, J. M.: N-gram-based text categorization. In.: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US, P. 161–175 (1994)
3. Vatanen, T, Väyrynen, J.J., Virpioja, S.: Language identification of short text segments with n-gram models. In.: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), P. 3423–3430 (2010)
4. Gusev, G., Chepovskiy, A.: The model for identification of a natural language of the text. Business Informatics, 3(17), P. 31–35 (2011)