Adv Syst Sci Appl 2018; **1**; 1-11
Published online at http://ijassa.ipu.ru.

Algorithms for prosodic discourse feature interpretation in case of its processing using low-speed codecs.

Maxim A. Bessonov[1], Natalia A. Bessonova[2], Mais P. Farkhadov[3]

[1)] *Peoples' Friendship University of Russia, Moscow, Russia*

E-mail: bessonovma@gmail.com

[2)] *Peoples' Friendship University of Russia, Moscow, Russia*

E-mail: nbesson@yandex.ru

[3)] *V.A. Trapeznikov Institute for Control Sciences of Russian Academy of Science, Moscow, Russia*

E-mail: mais.farhadov@gmail.com

Abstract: In this article we propose two algorithms for discourse prosodic feature interpretation. The first algorithm based on wide phonetic categories and second algorithm based on audio signal melodic cross-correlation functions and short-timed energy series – as well as methodical recommendations for their use are proposed as a part of the problem of audio signal language identification based on a prosodic approach. An experimental evaluation of both algorithms is proposed. Neural networks are used as a decision rule. Wide phonetic categories were pause, pitch, noise. We have expanded wide phonetic categories to pause, pitch, noise, five levels of pitch, sites of decreasing energy, main maximum, adverse maximum. The total number of categories was 14. These algorithms can be applied for language identification or speaker identification. At the same time there is no requirement to restore the speech signal after processing it by low-speed codec. Certainly, frames of the speech codec must contain such parameters as pitch, tone-noise parameter, energy. The base of speech signals consists of 10 languages 10 speakers per language. Total time of the speech per speaker is 100 minutes. This time takes into account statistical regularities of languages. Tests for evaluation of the algorithms were carried out with a multilayer perceptron.

Keywords: language identification, neural networks, discourse prosodic feature, wide phonetic categories.

1. INTRODUCTION

As numerous on-line human-machine interfaces are being developed, the problem of language identification stays unsolved. Moreover, those systems are often required to support numerous languages. There are four methods for language identification: acoustic, phonotactic, lexical and prosodic. In one or more of their aspects, the first three methods are based on the discourse signal parameters: acoustic, mel-frequency cepstral coefficients, mixed mel-frequency cepstral coefficients and others. The prosodic approach uses parameters such as discourse melody, rhythm, tone and others [5–9]. Prosodic parameters are difficult to describe and to interpret. Therefore, in the present article two algorithms are proposed for discourse prosodic feature description in order to use them in automatic audio signal language assessment systems. The first algorithm is based on wide phonetic categories [4]. The second is based on discourse melody cross-correllation function and short-timed energy series.

The main difference between the proposed algorithms and the ones described in the literature lays in their utilization for language assessment of audio signals that have been processed using low-speed codecs. This is based on the fact that low-speed codecs transmit

in communication channels such parameters as basal tone frequency, tone-noise signal and amplification of quasi-periodic fragments.

2. ALGORITHMS FOR PROSODIC FEATURES INTERPRETATION

2.1. Algorithm based on wide phonetic categories

Let $L=\{L_1,L_2,\ldots,L_N\}$ be an ensemble of languages on which the language assessment procedure is performed, with N being the overall language number. Let every language be represented by an ensemble $L_i=\{l_1,l_2,\ldots,l_{Mi}\}$ of audio recordings from different speakers of this specific language, with M_i being the overall number of audio recordings for a given language L_i.

An audio recording is divided in quasi-stationary segments $s_i(m)$, each one of which having a duration of K samples, with i being the segment number of a given discourse signal: $i = 1, 2, \ldots, P$, and P being the overall number of segments for a given discourse audio signal recording $m=1,\ldots,K\text{-}1$. Features are computed on each segment i, depending on its nature: vocal, non vocal or break

$$A_i = T\left(s_i\left(m\right)\right), i = 1, 2, ..., P \tag{2.1}$$

T being the operation allowing to define the type of segment. The evaluation of the segment's short-timed energy is denoted as

$$E_{k_i} = E\left(s_i\left(m\right)\right), i = 1, 2, ..., P \tag{2.2}$$

with E being the computation of the segment's short-timed energy. If the algorithm is used without reconstructing the source vocal signal waveform, then the parameters A_i and E_{k_i} are computed from the vocal transmission. Thus series $\vec{A} = \left(A_1, A_2, ..., A_P\right)$and $\vec{E}_k = \left(E_{k1}, E_{k2}, ..., E_{kP}\right)$ are formed. If the segment is classified as a break, then A_i=0. If the segment is classified as non-vocal, then A_i=1. For each vocal segment the basal tone frequency (BTF) is computed as

$$F0_i = F\left(s_i\left(m\right)\right), i = 1, 2, ..., P \tag{2.3}$$

with F being the operation of basal tone computation. Afterwards the series $\overrightarrow{F0} = \left(F0_1, F0_2, ..., F0_P\right)$ are formed. If the algorithm is used without reconstructing the source vocal signal waveform, the parameter $F0_i$ is computed from the vocal transmission. The basal tone frequency's range of variations is then divided into 5 intervals. Each vocal segment is attributed a number, depending on which BTF interval its frequency corresponds to

$$F0_{u_i} = UF\left(\overrightarrow{F0}\right), i = 1, 2, ..., P \tag{2.4}$$

$F0u_i$ being the BTF level, UF the operation of BTF change computation and segment encoding with numerical values $\overrightarrow{F0u} = \left(F0u_1, F0u_2, ..., F0u_P\right)$. This allows the formation of BTF values series for audio signal segments. Afterwards segments during which the discourse short-timed energy increases or decreases are computed as

$$E_{u_i} = UE\left(\overrightarrow{E_k}\right), i = 1, 2, ..., P \tag{2.5}$$

The encoding Eu_i=(+/–)1 depends on whether the energy variation is increasing or decreasing correspondingly, UE being the operation of audio signal short-timed energy computation. The series $\overrightarrow{Eu} = \left(Eu_1, Eu_2, ..., Eu_P\right)$are formed. If the short-timed energy decreases for a given segment, its BTF value is multiplied by (-1).

Principal and lateral BTF maxima on a segment between two breaks are used in order to

assess the principal and lateral accents. If the BTF and short-timed energy maxima correspond in time and are maximum for a given fragment, then this segment is considered a principal maximum. If the maxima do not correspond in time, then the fragment is considered a lateral maximum $MAX_i = \Theta(\overrightarrow{F0u}, \overrightarrow{Eu})$ with Θ being the operation of principal and lateral maxima computation for the BTF and short-timed energy series. This allows the constitution of the series

$$\overrightarrow{MAX} = (MAX_1, MAX_2, ..., MAX_P) \tag{2.6}$$

Therefore the final series $\overrightarrow{X} = (X_1, X_2, ..., X_P)$ of wide phonetic categories for a given audio record is constituted by elements X_i, where

$$X_i = \begin{cases} 0, \text{ if } A_i - \text{break}, \\ 1, \text{ if } A_i - \text{non-vocal}, \\ 2, \text{if } F0_{u_i} - \text{interval } 1, \\ -2, \text{ if } F0_{u_i} - \text{interval } 1, E_{u_i} = -1, \\ 3, \text{ if } F0_{u_i} - \text{interval } 2, \\ -3, \text{ if } F0_{u_i} - \text{interval } 2, E_{u_i} = -1, \\ 4, \text{ if } F0_{u_i} - \text{interval } 3, \\ -4, \text{ if } F0_{u_i} - \text{interval } 3, E_{u_i} = -1, \\ 5, \text{ if } F0_{u_i} - \text{interval } 4, \\ -5, \text{ if } F0_{u_i} - \text{interval } 4, E_{u_i} = -1, \\ 6, \text{ if } F0_{u_i} - \text{interval } 5, \\ -6, \text{ if } F0_{u_i} - \text{interval } 5, E_{u_i} = -1, \\ 7, \text{ if } MAX_i - \text{lateral maximum}, \\ 8, \text{ if } MAX_i - \text{principal maximum}. \end{cases} \tag{2.7}$$

Figures 2.1 and 2.2 show the algorithm diagram that implements the discourse signal encoding.

The autocorrelation function $\vec{R} = \Psi(\overrightarrow{X})$ is then computed on the series of wide phonetic categories $\overrightarrow{X}$, with Ψ being the operation of autocorrelation function computation.

If the algorithm is used without reconstructing the source vocal signal waveform, the BTF values are computed from the vocal transmission. If the algorithm is used with the source vocal signal waveform being reconstructed, then an algorithm for BTF evaluation is required.

There are numerous algorithms for BTF evaluation [2]. This article presents the comparison of already implemented algorithms: the algorithm SIFT, based on the autocorrelation function; the algorithm AMDF, based on short-time average difference function; and the algorithm for BTF evaluation of the MELP language encoding algorithm. Table 2.1 shows the percentage values for correct BTF computation *P*(OT), erroneous assumption that a vocal fragment is non-vocal *P*(HB/B) and erroneous assumption that a non-vocal fragment is vocal *P*(B/HB).

Table 2.1. Evaluation of BTF correct evaluation

Algorithm	SIFT	AMDF	MELP
P(OT), %	87±1	89±1	95±1,5
P(HB/B), %	7±1	6±1	3±0,5
P(B/HB), %	0,5	0,5	0,5

The algorithm MELP was used for BTF computation as its experimental evaluation proved it to be the most effective.

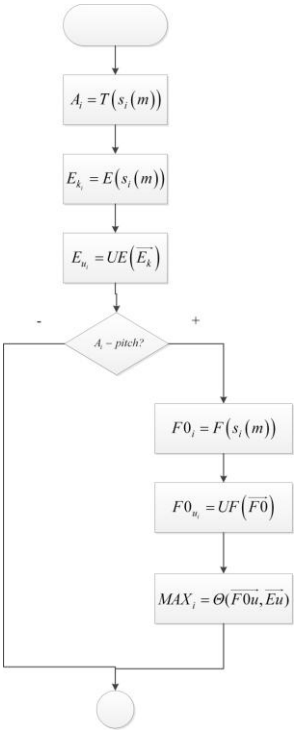

Fig. 2.1. Algorithm diagram for vocal signal segments encoding

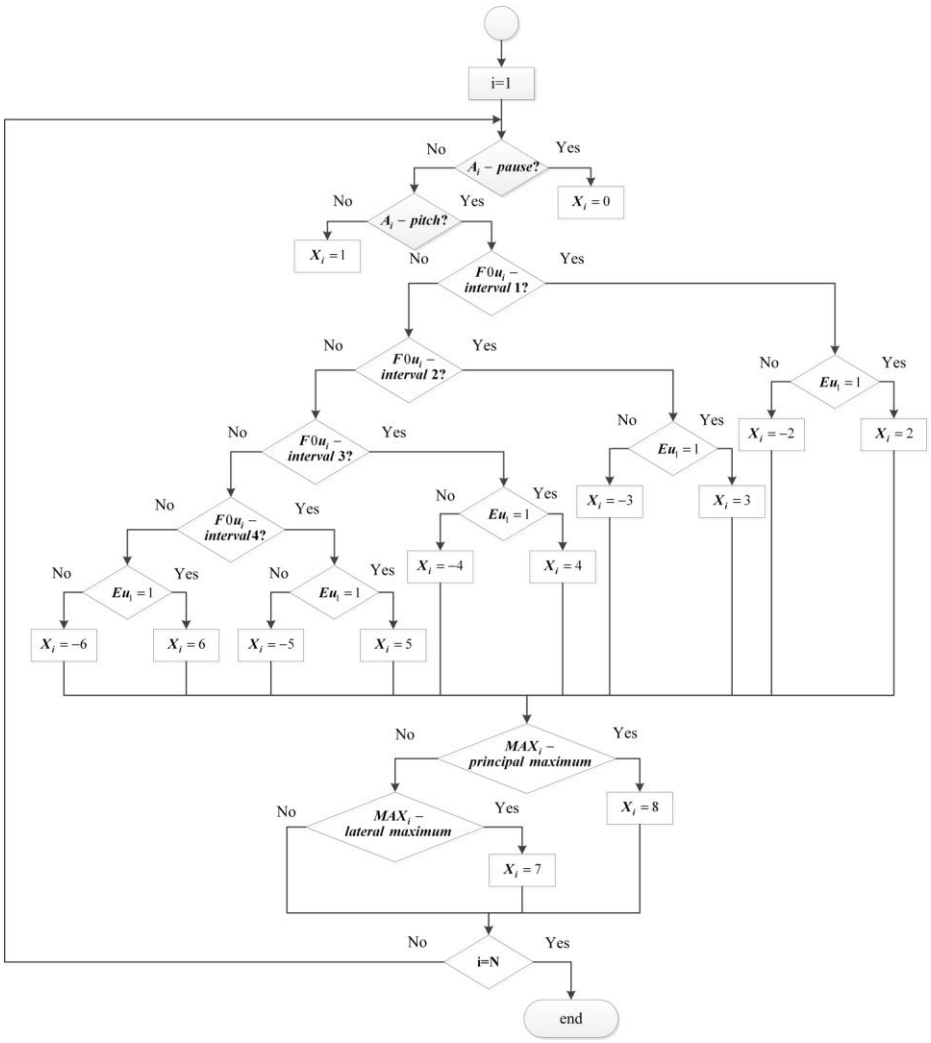

Fig. 2.2. Algorithm diagram for vocal signal segments encoding (continued)

2.2. Algorithm based on basal tone melody cross-correlation functions and short-timed energy series

The prosodic classification may be realized using basal tone melody cross-correlation function and short-timed energy of audio recordings. Each audio recording is divided in quasi-stationary segments $s_i(m)$ of K samples where i is the vocal signal number: $i = 1, 2, \ldots, P$, P is the overall number of segments in a vocal signal $m=1,\ldots,K$-1. For each segment i features are computed in accordance to the segment's nature: vocal, non vocal or break

$$A_i = T\left(s_i\left(m\right)\right), i = 1, 2, ..., P \tag{2.8}$$

T being the operation of segment type computation and the computation of the short-timed energy for one segment being

$$E_{k_i} = E\left(s_i\left(m\right)\right), i = 1, 2, ..., P \tag{2.9}$$

E being the operation of short-timed energy computation. The series $\vec{A} = \left(A_1, A_2, ..., A_P\right)$ and $\vec{E}k = \left(Ek_1, Ek_2, ..., Ek_P\right)$ are formed correspondingly. If a segment is classified as a break then A_i=0. If a segment is classified as non-vocal, then A_i=1. For each vocal segment the BTF is computed

$$F0_i = F\left(s_i\left(m\right)\right), i = 1, 2, ..., P \tag{2.10}$$

F being the operation of BTF computation. Afterwards series $\overrightarrow{F0} = \left(F0_1, F0_2, ..., F0_P\right)$ are formed. If the algorithm is used without reconstructing the source vocal signal waveform, the

parameter A_i, Ek_i, and $F0i$ is computed from the vocal transmission.

The cross-correlation function is computed on the BTF series and the short-timed energy series

$$\vec{B} = \Phi(\overrightarrow{F0}, \overrightarrow{Ek}) \tag{2.11}$$

Φ being the operation of BTF melody cross-correlation function computation and short-timed energy series. The vector formed by the cross-correlation function values and series of wide phonetic categories is then given at the first layer of the neural network, that is used to infer the language group to which the presented vector corresponds.

The feature computation algorithm is presented in Figure 2.3.

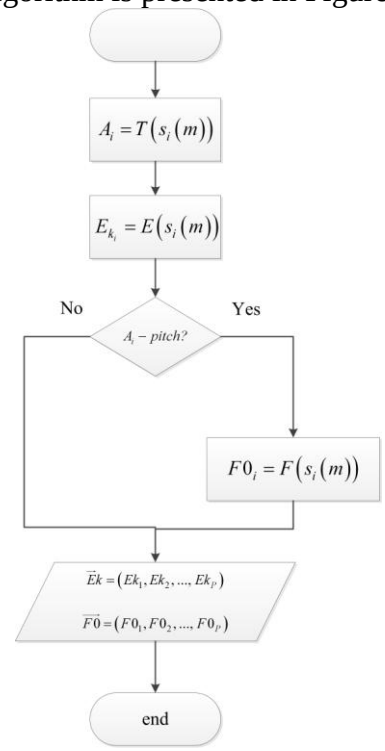

Fig. 2.3. Diagram of the algorithm of vocal segment encoding

3. METHODICAL RECOMMENDATIONS FOR THE USE OF ALGORITHMS FOR PROSODIC FEATURE INTERPRETATION ALGORITHMS AS A PART OF THE AUDIO SIGNAL LANGUAGE IDENTIFICATION

Methodical recommendations were developed in order to apply the aforementioned algorithm. They contain a succession of phases.

Phase 1. Discourse dataset formation for training. The training dataset must fulfill the following conditions: if N is the overall number of languages, d_m^i the number of male speakers for a given language i, d_m^i the number of female speakers for a given language i, then $V_i(d_m^i, d_f^i) = V_j(d_m^j, d_f^j)$, where i, j are the numbers of languages $i, j = 1, \ldots, N$, meaning that all age groups must be represented in equal proportion among male and female speakers or that the volume of voice data for all age groups must be equal. The volume of voice data must be sufficient from a statistical standpoint in order for all the pronunciation variations to be described. The overall data volumes must be equal for every languages.

Step 1. Reception from the source of a digital signal under the form $S_t(f_d, m, p, f_r)$, having the following characteristics: format "wav", sampling frequency f_d = 8kHz, regime m = mono, datadepth p = 16 bits, t being the audio signal number.

Step 2. Filtering of the audio signal $S_t(f_d, m, p, f_r)$ for unwanted noise suppression. This allows to receive the filtered signal $S_t^f(f_d, m, p, f_r) = P[S_t(f_d, m, p, f_r)]$, where P is the filtering operation.

Step 3. Training and control dataset formation. A set of audio signals $Z_{Li}\{S_1^f{}_{Li}(f_d, m, p, f_r), S_2^f{}_{Li}(f_d, m, p, f_r), \ldots, S_{Mi}^f{}_{Li}(f_d, m, p, f_r)\}$ is formed for each language L_i, where Mi is the overall number of audio signals for a given language Li. The complete set of audio signals is then denoted by $Z = \{Z_{L1}, Z_{L2}, \ldots, Z_{LN}\}$.

Step 4. Treatment of all audio signals in every language made available by the vocal transmission. $Z^{vok} = \text{VOK}(Z)$, VOK where denotes the processing of the voice transmission dataset , $Z^{vok} = \{Z^{vok}{}_{L1}, Z^{vok}{}_{L2}, \ldots, Z^{vok}{}_{LN}\}$.

Step 5. Audio signal parameter computation using the presented algorithms. Afterwards a set of parameters $Z^{vok}{}_{Li}^{Mod1} = \text{Mod1}(Z^{vok}{}_{Li})$, $Z^{vok}{}_{Li}^{Mod2} = \text{Mod2}(Z^{vok}{}_{Li})$ is created, where Mod1 and Mod2 are the operation of parameter computation using the presented algorithms for prosodic parameters description.

Phase 2. Neural network training. The neural network training operations allow the fine tuning of its parameters. Neural networks with different topologies are described by different mathematical models, therefore in every specific situation a neural network is described by a specific formula. Neural networks are defined for language groups. Their number is equal to the number of combinations of 2 elements out of N.

Phase 3. Neural network performance evaluation

Step 1. Reception from the source of a digital signal under the form $S_t(f_d, m, p, f_r)$, having the following characteristics: format "wav", sampling frequency f_d = 8kHz, regime m = mono, data depth p = 16 bits, t being the audio signal number.

Step 2. Filtering of the audio signal $S_t(f_d, m, p, f_r)$ for unwanted noise suppression. This allows to receive the filtered signal $S_t^f(f_d, m, p, f_r) = P[S_t(f_d, m, p, f_r)]$, where P is the filtration operation.

Step 3. Neural network's testing. At the neural network's input, for each language pair L_i and L_j, audio signals in the language i and j are given. The neural network's output gives an evaluation of the audio signal's language identity, given t: the audio signal's number:

$$\hat{L}_t = NET\left(S_t^f\left(f_d, m, p, f_r\right)\right) \tag{3.1}$$

Step 4. Evaluation of the number of correctly assessed audio signal for each language pair. This forms the vector $D = (d_{12}, d_{21}, d_{13}, d_{31}, \ldots, d_{N(N-1)}, d_{(N-1)N})$, where d_{ij} is the number of correctly assessed audio signal for a given language pair L_iL_j, $i \neq j$.

Step 5. Hierarchical language tree construction based on the agglomerative hierarchical algorithm

$$\rho_{min}\left(\omega_i, \omega_j\right) = \min_{x_k \in \omega_i, x_l \in \omega_j} d\left(X_k, X_l\right) \tag{3.2}$$

where ω_i and ω_j are the languages L_i and L_j, and $\rho(\omega_i,\omega_j)$ is the distance between L_i and L_j. The hierarchical language tree is the base upon which language groups are formed.

4. DISCOURSE DATABASE FORMATION

An audio signal dataset was formed in order to conduct test according to the presented methodical recommendation. Its content is summarized in the Table 4.1.

Audio records were taken from internet translation resources: television and radio, which implies that the discourse were processed using different codecs.

In order to exclude the influence of the dataset's constitution on the experiment results, the number of speakers in each language was chosen equal. The overall duration of the audio signals was as well equal. The dataset was split equally in training and validation subsets. Samples from the validation set were not present in the training set. For experimental purposes, all audio records both in the training and validation datasets were divided in 10-seconds long fragments.

The speakers' age repartition was approximated: men and women aging from 20 to 50 years old. 80% of each speaker's audio signal time was used for training, 20% for validation. The separation into training and validation was performed randomly.

Table 4.1. Contents of the dataset used for prosodic feature models experimental validation

Language	Number of speakers	Overall audio signal duration for each speaker, min	Speaker sex (m-male, f-female)	Repartition in the training/testing sets, %
Chinese	10	100	5m/5f	80/20
English	10	100	5m/5f	80/20
Finnish	10	100	5m/5f	80/20
French	10	100	5m/5f	80/20
German	10	100	5m/5f	80/20
Japanese	10	100	5m/5f	80/20
Farsi	10	100	5m/5f	80/20
Portuguese	10	100	5m/5f	80/20
Russian	10	100	5m/5f	80/20
Spanish	10	100	5m/5f	80/20

5. NEURAL NETWORK DEFINITION AND TUNING

Pattern recognition tasks are in most cases solved using statistical methods. However in the case of vocal data in different languages, it proves difficult to build the repartition function of the considered parameters. Therefore in the present article neural networks were used for vocal segments classification.

As stated in literature, for classification-type problems, the number of neurons in the network's first layer is equal to the number of elements in the feature vector that is given at the entrance [3]. The number of output neurons depends on the type of problem and the output number interpretation rules [3]. The number of neuron in the intermediate layers is given by the formula [3]

$$\frac{N_y N_p}{1+log_2\left(N_p\right)} \le N_w \le N_y\left(\frac{N_p}{N_x}+1\right)\left(N_x+N_y+1\right)+N_y \tag{3.3}$$

where N_y is the neural network's (NN) output vector's number of element, N_p is the number of elements in the test dataset, N_x is the number of element in the input vector and N_w is the overall number of neurons.

The choice of the NN's class and architecture is a non-trivial problem for which exact solutions do not exist [3]. In order to choose the number of neurons, one can highlight two methods: the more neurons, the more reliable the network will be and the more neurons, the worse the network will approximate the transfer function. The neural networks were implemented in MATLAB, using the environment's built-in functions.

The following architectures were experimentally evaluated: Kohonen maps, cascade-forward NN, Elman networks, multilayer perceptron, Hopefield networks, probabilistic networks, networks with Radial Basis Functions (RBF), counter-propagation network with Learning Vector Quantization.

The networks were trained with built-in MATLAB functions [1]: quasi-Newton algorithm, Levenberg-Marquardt algorithm with Bayes regularization, Fletcher-Reeves conjugate gradient method, Polak-Ribiére conjugate gradient method, Powell-Beale conjugate gradient method, gradient descent, gradient descent with variable learning rate, Levenberg-Marquards algorithm, scaled conjugate gradient method, gradient descent with momentum, gradient descent with momentum and variable learning rate, one step secant method, random increment method and elastic error backpropagation algorithm.

For the first phase, in order to build the limited groups of 10 languages, experiments were performed with each separated language pair, accounting for 45 neural networks in total.

The best results have been obtained using a multilayer perceptron. Therefore this architecture has been selected for fine tuning.

Since the language given at the NN's entrance is a priori unknown, it was decided to use a

unified architecture for each language pair.

6. EVALUATION OF THE WIDE PHONETIC CATEGORIES ALGORITHM

According to the formula and the starting conditions for NN testing $N_y = 2$, $N_p = 600$, the number of neurons in the hidden layers is then $117 \le N_w \le 2015$ for the wide phonetic categories model.

Since *Nw* is between 117 and 2015, at the moment of the NN architecture definition the number of neuron was chosen from 100 to 2000, correspondingly to the number of neurons from 1 (one layer from 100 to 2000 neurons) to 20 (20 layers of 100 neurons) in the following configurations: from 100 to 1000 neurons with a 10 neuron step in a given layer or from 1000 to 2000 with a 100 neuron step. The maximal number of neurons in one given layer was 800.

In order to build the different multilayer perceptron architectures for the 45 language pairs, a vector $D=(d_{1,2}, d_{2,1}, d_{1,3}, d_{3,1}, d_{i,j}, d_{j,i}, d_{N,N-1}, d_{N-1,N})$ of goal indicators for assessment confidence was built, with N being the overall number of languages in the Automatic Language Assessment System. Therefore the length of the vector is D=90. Each element $d_{i,j}$, $d_{j,i} = 100$.

The vector $D_k=(d^k_{1,2}, d^k_{2,1}, d^k_{1,3}, d^k_{3,1}, d^k_{i,j}, d^k_{j,i}, d^k_{N,N-1}, d^k_{N-1,N})$ of goal indicators for assessment confidence for the current architecture has as well 90 elements. The distance between D and D_k is defined as

$$D_r = \frac{\sqrt{\left(d_{1,2} - d^k{}_{1,2}\right)^2 + \left(d_{2,1} - d^k{}_{2,1}\right)^2 + \left(d_{i,j} - d^k{}_{i,j}\right)^2 + \ldots}}{\ldots + \left(d_{j,i} - d^k{}_{j,i}\right)^2 \ldots + \left(d_{N,N-1} - d^k{}_{N,N-1}\right)^2 + \left(d_{N-1,N} - d^k{}_{N-1,N}\right)^2} \tag{3.3}$$

Thus, the lower the distance D_r, the better the NN tuning. At the end it was found that D_r lays in the interval from 59.1861 to 532.4106. The best value, $D_r = 72.5358$, was obtained for a NN with 1400 neurons overall, organized in one layer of 800 neurons and 2 layers of 600 neurons. The results of language assessment are presented in the Table 6.1.

Table 6.1. Average confidence values for language identification

	Chinese	English	Finnish	French	German	Japanese	Persian	Portuguese	Russian	Spanish
Chinese		94.5	95.1	96.2	95.9	97.5	96.6	95.2	94.4	97.9
English	93.8		97.4	92.8	93.8	93.6	98.1	94.5	94.0	97.8
Finnish	93.8	93.7		93.2	93.4	93.9	93.9	96.1	93.7	94.3
French	94.2	93.6	93.2		93.9	93.4	94.0	94.8	93.8	94.4
German	94.5	92.6	93.7	92.5		94.6	94.0	97.5	96.3	93.9
Japanese	83.6	94.1	74.0	98.3	93.3		94.0	84.9	94.4	98.0
Persian	84.4	94.0	74.6	93.3	93.8	83.6		92.7	84.3	93.2
Portuguese	94.2	93.6	93.5	93.9	94.2	94.5	93.5		93.9	98.4
Russian	94.4	95.1	94.1	95.3	93.4	94.0	94.4	94.3		94.5
Spanish	93.9	94.3	93.4	93.2	94.2	93.8	94.1	94.5	93.2	

The languages were used to form groups using the agglomerative algorithm. Language pairs were used in quality of patterns to be recognized. The average confidence value for fixed first and second order error rates was used as a measurement of the distance between the two languages of one pair. The distance between classes is defined by the distance to the nearest neighbor:

$$\rho_{min}\left(\omega_i, \omega_j\right) = \min_{x_k \in \omega_i, x_l \in \omega_j} d\left(X_k, X_l\right) \tag{3.5}$$

where ω_i,ω_j are the languages L_i and L_j, and $\rho(\omega_i,\omega_j)$ is the distance between L_i and L_j.

This allows the creation of a graph of hierarchical classification. This graph can then be used to assess groups of languages close to each other.

7. EVALUATION OF THE BASAL TONE MELODY CROSS-CORRELATION FUNCTION AND SHORT-TIMED ENERGY SERIES ALGORITHM

According to the formula and the NN test starting conditions, $N_y = 2$, $N_p = 600$, $N_x = 797$, the number of neurons in the hidden layer is $117 \le N_w \le 2806$ for the cross-correlation function and BTF and short-timed energy series model.

Since N_w is comprised between 117 and 2086, at the time of the NN architecture definition the number of neurons in a layer was varied from 100 to 3000, correspondingly layers having from 1 (1 layer from 100 to 3000 neurons) to 20 (30 layers of 100 neurons each). The following configurations were used: from 100 to 1000 neurons with a 10 neuron step by layer, from 1000 to 3000 with a 100 neuron step. The maximum number of neurons was 800, $D_r = 89.1449$.

The results of language identification are presented in the Table 7.1.

Table 7.1. Average confidence values for language

	Chinese	English	Finnish	French	German	Japanese	Persian	Portu-guese	Russian	Spanish
Chinese		97.7	94.7	92.8	97.8	97.9	93.8	91.7	93.1	92.1
English	91.2		91.4	92.3	92.9	94.8	92.7	97.7	90.3	92.0
Finnish	90.9	91.5		95.8	94.7	94.6	95.4	90.9	93.6	95.9
French	92.1	92.9	92.4		93.9	96.7	97.5	92.1	91.8	91.8
German	92.5	90.2	91.4	90.4		91.8	92.2	92.4	93.0	95.4
Japanese	80.6	91.8	90.1	82.3	71.9		90.7	90.5	94.7	97.2
Persian	71.1	91.5	82.3	91.6	82.6	78.2		97.5	92.5	91.5
Portuguese	90.7	91.0	92.0	92.0	93.2	93.4	92.1		94.5	92.5
Russian	91.0	91.7	90.6	92.6	92.3	92.4	91.7	91.6		96.2
Spanish	90.5	92.9	90.9	92.8	91.2	93.1	91.4	92.1	93.6	

8. CONCLUSION

The algorithms presented in the article aim at a complex description of discourse prosodic feature for their usage in special data processing tasks, in particular audio signal language assessment. As seen in the presented tables, prosodic feature description using wide phonetic categories allows for high-confidence language identification. However this performance insignificantly surpasses cross-correlation function. Distance indicators for current results of language assessment with respect to the goal indicator D_r scored at $D_r = 72.5358$ for the autocorrelation model from wide phonetic categories and $D_r = 89.1449$ for the signal's cross-correlation function model from basal tone value and short-timed energy series.

The presented algorithms demark themselves from others in that they are used for audio signal language identification, after vocal transmission, but without reconstructing the source vocal signal waveform.

REFERENCES

1. Diakonov V. & Kruglov V. (2006) *Matlab 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6.Instrumenti iskusstvennogo intellekta i bioinformatiki* [Matlab 6.5 SP1/7/7 SP1/7

SP2 + Simulink 5/6. Tools of artificial intelligence and bioinformatics]. Moscow, Russia: Solon-press [in Russian].

2. Imamverdiev Y. & Suhostat L. (2014) Podhodi dlia ozenki perioda osnovnogo tona rechevogo signala v zashumlennoy srede [The approaches for pitch evaluation in noisy environment]. *Speech technology.*, 4: 84-102.
3. Komartsova L. & Maksimov A. (2004) *Neirokimpiuteri: Uchebnoe posobie* [Neurocomputers: tutorial]. Moscow, Russia: BMSTU [in Russian].
4. Miloshenko A.A. (2010) *Razrabotka metodiki ispolzovania shirokih foneticheskih kategoriy v zadachah verifikazii diktora* [Development of method of using broad phonetic categories in speaker identification task]. Moscow, Russia: MIIT [in Russian].
5. Ambikairajah E., LI H., Wang L. & Yin B. (2011) Language Identification: A Tutorial. *IEEE Circuits and Systems Magazine*. 11(2), 82–108.
6. Bhattacharjee U. & Sarmah K. (2013) Language identification system using MFCC and prosodic features. *Int. Conference on Intelligent Systems and Signal Processing (ISSP)*, Gujarat, 194–197.
7. Lee R., Leung C.-C. & Ma B. (2013) Spoken Language Recognition with prosodic features. *IEEE Trans. on Audio, Speech, and Language Processing*. 21(9), 1841–1853.
8. Martinez D., Jeida E. & Ortega A. (2013) Prosodic features and formant modeling for an iVector-based language recognition system. *IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, 6847–6851.
9. Martínez D., Burget L., Ferrer L. & Scheffer N. (2012) iVector-based prosodic system for language identification. *IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 4861–4864.

