Peculiarities of matching the text and sound components in the Ukrainian language system development

Taras Basyuk[1,*,†], Andrii Vasyliuk[1,*,†]

[1] *Lviv Polytechnic National University, Bandera str.12, 79013, Lviv, Ukraine*

Abstract

This article analyzes the existing methods and known systems that provide tools for recognizing Ukrainian language and describes approaches and methods for synchronizing text and audio information. The relevance of creating a system is proved, and the prefaces of scientific research in this area are described. To present the main aspects of the studied subject area, the classification of sounds in Ukrainian language was considered, and the features of their detection and formation were given. The next stage was to determine the study of spectral analysis and its influence on the recognition process. Namely, it has shown an influence on the acoustic features selection of speech, which subsequently made it possible to determine the sequence of phonemes that correspond to the input signal. The stage of the audio stream and phoneme units' synchronization, using the GMM algorithm, is described. The main idea was to build a model of the audio stream that can be compared with vectors of phonemic features to determine the correspondence between them. The mathematical description of the specified process is performed using algebra of algorithms. An applied software system has been developed that implements text and audio information synchronization. The main stages of the system work: text analysis, transcription creation, spectral analysis of the audio track, search for phoneme characteristics in the audio track, application of the GMM algorithm and output of results. At the current moment, the software solution works in the form of a prototype.

Further research will be directed to testing and improving the system, eliminating conflicts, and expanding functionality in accordance with the specified requirements.

Keywords

ukrainian-language content, speech recognition, text analysis, GMM algorithm, transcription

1. Introduction

The development of a system for matching the text and audio components of Ukrainian-language content is of great relevance in the modern information environment. It allows you to improve the user experience of audio content consumers, improve the efficiency of voice interfaces, and ensure content accessibility for people with disabilities. Technological progress in natural language processing and artificial intelligence makes the development of such

MoDaST-2024: 6th International Workshop on Modern Data Science Technologies, May, 31 - June, 1, 2024, Lviv-Shatsk, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

Taras.M.Basyuk@lpnu.ua (T. Basyuk); Andrii.S.Vasyliuk@lpnu.ua (A. Vasyliuk)

0000-0003-0813-0785 (T. Basyuk); 0000-0002-3666-7232 (A. Vasyliuk)

systems more effective and promising. In general, the process consists of two stages: speech recognition and comparison of the received content with the existing textual component [1, 2].

Speech recognition is a technology that allows a computer to identify individual words or phrases spoken by a person and convert them into text. This field includes knowledge and research in computer science, linguistics, and electrical engineering. Speech recognition systems are gradually becoming an intermediary between humans and technological devices, providing alternative methods of information exchange. Along with software for dictation on personal computers, more advanced systems are being developed, such as voice assistants (Siri, Google Assistant, Alexa, Cortana), which, in addition to executing commands, can conduct a live dialogue and solve applied problems. However, most of them require access to the Internet, which limits their use, and the speed of operation depends on the quality of the Internet connection [3, 4]. It is important to note that most such systems do not support the Ukrainian language due to its specificities, such as high inflection and free order of words in a phrase or sentence. This leads to difficulties in recognition and reduces the accuracy of work. Therefore, it is necessary to look for new methods and algorithms for recognizing the Ukrainian language and adapt them to solve the given task.

To date, various approaches have been tested to recognize the words of fused speech [5-7]. In the first case, with the global approach, the word to be recognized is compared with every word in the dictionary. When comparing, as a rule, the spectral representation of each word is used [8]. Among various methods of this type, the dynamic programming method gave satisfactory results [9]. In the second case, with an analytical approach, each word or group of words is first segmented into smaller units. This allows you to perform recognition at the syllable or the phoneme level and store in memory the parameters (duration, energy, etc.) associated with the event. Segmentation can be based on finding vowels, which are often located near the maximum of the integrative energy of the spectrum. With this approach, the first criterion for segmentation is the change in energy over time [10].

In view of the above information, an urgent task is to develop a system for matching the text and audio components of Ukrainian-language content, which will provide means of effective recognition and reproduction of content in Ukrainian-language fusion broadcasting.

1.1. Analysis of recent researches and publications

1.1.1. Analysis of automatic speech recognition stages

The problem of automatic speech recognition can be solved step by step. At the first stage, the task of recognition consists in the external search for characteristics and only superficially characterized classes of acoustic events. For the second stage, the generalization of external criteria for the classification of internally undetected classes is crucial, which makes it possible to predict the characteristics of an unknown signal [11]. In automatic speech recognition, firstly, it is necessary to find out whether the signal is phonetic (speech) [12].

It is known about the division of the speech flow into micro and macro segments. The distinction between two macro-segments (phrases, syntagms) is, as a rule, discrete, and between two micro-segments (sub sounds, sounds, syllables) is blurred. Sounds change their suprasegmental (duration, intensity, frequency of the fundamental tone) and segmental (spectral) characteristics according to the influence of other parameters. For example, an increase in the duration of a vowel component in a speech stream may indicate semantically highlighted words, etc. Therefore, to predict, for example, the duration of a sound, several linguistic factors should be considered [13].

Here we should dwell on some segmentation problems related to the specificity of the phonetic level. Automatic recognition of nasal and smooth phonemes of fused speech can be included among the difficulties [10]. Uncertainties arising from the limitations of any language processing system and often due to poor pronunciation are considered as sources of information for stochastic or uncertain set grammar [14, 15]. Currently, available methods of micro segmentation of speech (segmentation into sub sounds, sounds, syllables) are classified as follows [16, 17]:

1. Using the degree of stability over time of any acoustic parameters of the speech signal, such as the concentration of energy in the frequency spectrum.
2. Superimposition of acoustic labels on the speech signal at regularly repeated short intervals.
3. Comparison of speech signal samples in abbreviated time windows at regular intervals with samples from phoneme prototypes.

There are context-dependent and context-independent methods of segmentation. The simplest method of context-independent marking is comparison of standards [18]. This requires that the device stores a model for each vocabulary item. Context-dependent segmentation allows the connection of a set of features use and thresholds with the phonetic context. Usually, the task of speech recognition is reduced to the task of recognizing individual sounds with the subsequent use of algorithms that consider the peculiarities of pronunciation, word formation and phrasing of some individuals [19].

In this case, the task of distinguishing speech sounds can be considered as a task of pattern recognition, the number of which is limited, although it reaches several dozen. At the same time, classifying the proposed sound samples can be reduced to multi-alternative hypothesis testing. At the same time, the speech sound recognition system can be built using the principles of 'learning with a teacher' [20], that is, a preliminary set of information base of classified data with which comparison is made. The procedure for recognizing speech sounds should consider the peculiarities of their implementation. First, these realizations have their own appearance for each sound. Secondly, they have a limited length of time [21].

Speech signal analysis methods can be considered using a model in which the speech signal is the response of a system with slowly changing parameters to periodic or noise-exciting oscillations [22]. A speech signal can be modeled by the response of a linear system with variable parameters (vocal tract) to the corresponding excitatory signal. With an unchanged form of the vocal tract, the output signal is equal to the convolution of the excitatory signal and the impulse response of the vocal tract. However, all the variety of sounds is obtained by changing the shape of the vocal tract. If the shape of the vocal tract changes slowly, then at short time intervals, the output signal is logically approximated by the convolution of the excitatory signal and the impulse response of the vocal tract [23, 24]. Since the shape of the vocal tract changes when creating different sounds, the spectral envelope of the speech signal will also change over time. Similarly, when the period of the signal that excites ringing sounds changes, the frequency difference between the harmonics of the spectrum will change.

Therefore, in the process of recognition, it is necessary to know the type of speech signal in short periods of time and the nature of its change over time.

1.1.2. Analysis of speech signals

Having analyzed the stages of automatic speech recognition, we can conclude that speech signal analysis systems usually try to separate the excitatory function and the characteristics of the vocal tract. Next, depending on the specific method of analysis, the parameters describing each component are obtained [25]. In the frequency domain, the spectrum of short segments of the speech signal can be represented as the product of the contour characterizing the state of the vocal tract and the function characterizing the excitatory signal. Since the main parameter of the signal, an exciting ringing sound, is the spread of harmonics of the fundamental tone, and the characteristics of the vocal tract are determined with sufficient completeness by the formant frequencies, it is very convenient to proceed from the representation of speech in the frequency domain during the analysis. When creating different sounds, the vocal tract's shape, and the excitatory signal change, so the spectrum of the speech signal also changes. Therefore, the spectral representation of speech should be based on the short-term spectrum, which can be obtained from the Fourier transform [26].

Consider a discretized speech signal represented by the sequence s(n). Its short-time Fourier transform is defined as [27]:

$$\sum_{k=-\infty}^{\infty} s\,(k)h(n-k)e^{-j\omega k}$$

This expression describes the Fourier transform of the weighted segment of the speech oscillation, and the weighting function h(n) shifts in time.

Linear prediction is one of the most effective methods of speech signal analysis. This method becomes the most common when evaluating the main parameters of speech signals, such as the period of the main tone, formants, spectrum, and when abbreviating speech for its low-speed transmission and economical storage. The importance of the method is due to the high accuracy of the obtained estimates and the relative simplicity of the calculation [28].

The basic principle of the linear prediction method is that the current count of the speech signal can be approximated by a linear combination of previous counts. At the same time, the prediction coefficient is uniquely determined by the minimization of the mean square of the difference between the readings of the speech signal and its predicted values (at the final interval). Prediction coefficients are weights used in a linear combination. The linear prediction method can be used to reduce the volume of a digital speech signal [29].

The main goal of processing speech signals is to obtain the most convenient and compact representation of their content. The accuracy of the presentation is determined by the information that needs to be preserved or highlighted. For example, digital processing can be applied to determine whether this oscillation is a speech signal. Most speech processing methods are based on the idea that the properties of the speech signal slowly change over time. This assumption leads to short-term analysis methods, in which segments of the speech signal are extracted and processed as if they were short segments of individual sounds with distinct properties. In the general case, the energy function can be determined as follows [30]:

$$\sum_{\text{m = -∞}}^{\infty}[x(m)\omega(n-m)]^2$$

This expression can be rewritten in the form:

$$E_n = \sum_{m=-\infty}^{\infty} x^2(m)h(n-m), where\ h(n) = \omega^2(n)$$

The impulse response *h(n)* choice or the window forms the basis of the signal description using the energy function. To understand how the choice of time window affects the short-term energy function of the signal, suppose that *h(n)* is long enough and has a constant amplitude; the value of *E* will change slightly over time. Such a window is equivalent to a low-pass filter with a narrow bandwidth. The band of the low-pass filter should not be so narrow that the output signal is constant. A narrow window (short impulse response) is desirable to describe rapid amplitude changes, but too small a window width can lead to insufficient averaging and, therefore, insufficient smoothing of the energy function. The influence of the time window width on the accuracy of the short-term average value measurement (average energy) is determined by the dependence: if *N* (the width of the window) is insignificant (close to the period of the main tone and less), then En will change very quickly, according to the fine structure of the speech oscillation; if *N* is large (several periods of the main tone), then *En* will change slowly and will not adequately describe changes in the features of the speech signal [31].

This means that there is no single value of *N* that fully satisfies the listed requirements, since the period of the fundamental tone varies from 10 counts (at a sampling rate of 10 kHz) for high children's and female voices to 250 counts for extremely low male voices. The main purpose of *En* is that this value allows you to distinguish vocalized speech segments from non-vocalized ones. The value of the function of the short-term mean value of the signal for non-vocalized segments is significantly smaller than for vocalized ones.

A characteristic feature of the speech signal analysis method is binary quantization of the input speech signal [32]. The used mathematical model of the speech signal has the form:

$$S(t) = A(t) \cdot e^j \psi(t),$$

where *A(t)* is the law of change in the amplitude of the speech signal, *Y(t)* is the full phase function of the speech signal.

The law of the signal amplitude change is not a sufficiently informative parameter for evaluating a speech message since it is not constant for the same word or phrase uttered with different intonation and volume. The speech signal's full phase function is assumed to be the informative characteristic of the speech signal in this method. The full phase function of the speech signal is presented in the form of a Taylor series expansion [33]:

$$\Psi(t) = \frac{\Psi^{(0)}(t_0)}{0!}t^0 + \frac{\Psi^{(1)}(t_0)}{1!}t^1 + \frac{\Psi^{(2)}(t_0)}{2!}t^2 + \frac{\Psi^{(3)}(t_0)}{3!}t^3 + \dots$$

The specified expression can be rewritten as follows:

$$\Psi(t) = \mu_0 + \mu_1 t + \frac{\mu_2 t^2}{2} + \frac{\mu_3 t^3}{6} + \dots$$

The first three expansion coefficients are taken in the schedule. At the same time, the first coefficient m0, which is the initial phase of the speech signal, is taken equal to zero, due to insignificant informativeness. Then the complete phase function will be determined:

$$\Psi(t) = \mu_1 t + 0{,}5mu_2 t^2$$

where, *m1* is the decomposition coefficient, which is the average frequency of the speech signal, *m2* is the decomposition coefficient, which is the change in the frequency of the speech signal. After discretization, the complete phase function has the following form:

$$\Psi(i \cdot \Delta t) = \mu_1 \cdot (i \cdot \Delta t) + 0{,}5 \cdot \mu_2 \cdot (i \cdot \Delta t)^2$$

where i is the number of the current count in the discretized sequence, Δt is the discretization step.

1.1.3. Analysis of software products

Commercial programs for speech recognition appeared in the early nineties. They were usually used by people who, due to a hand injury, were unable to type many texts. These programs (e.g., Dragon NaturallySpeaking, VoiceNavigator) translated the user's voice into text, thus relieving his hands. The translation reliability of such programs was not high, but it gradually improved over the years.

The increase in the computing power of mobile devices has made it possible to create programs with the function of speech recognition for them as well. Among such programs, it is worth noting the Microsoft Voice Command application, which allows you to work with many programs using your voice. For example, you can enable music playback in the player or create a new document. Intelligent language solutions that allow automatic synthesis and recognition of human speech are the next step in the development of interactive voice systems (IVR) [34]. Using an interactive phone application is not a fad but a vital necessity. Reducing the load on contact center operators and secretaries, reducing labor costs, and increasing the productivity of service systems - these are just some of the advantages that prove the feasibility of such solutions.

The next step in speech recognition technologies can be considered the development of the so-called Silent Speech Interfaces (SSI). These speech processing systems are based on the acquisition and processing of speech signals at the early stage of articulation [35]. This stage of speech recognition development is caused by two significant shortcomings of modern recognition systems: excessive sensitivity to noise, and the need for clear and distinct speech when addressing the system. The approach based on SSI is to use new sensors that are not affected by noise as a supplement to the processed acoustic signals.

Today, there are two types of speech recognition systems - client-based and client-server. When using client-server technology, a voice command is entered on the user's device and transmitted via the Internet to a remote server, where it is processed and returned to the device in the form of a command (Google Voice, Vlingo, etc.). Due to the enormous number of server users, the recognition system receives a significant training base. The first option works on other mathematical algorithms and is rare (Speereo Software) - in this case, the command is entered on the user's device and processed on it. The advantage of processing 'on the client' is mobility, independence from the presence of communication and the operation of remote equipment. In particular, the system working 'on the client' seems more reliable, but is limited, at times, by the power of the device on the user's side [36].

Speech recognition systems can also be divided into announcer-oriented and announcer-independent. Speech recognition systems aimed at working with announcers, or announcer-oriented systems, aimed at recognizing and analyzing the speech of specific individuals or groups of announcers. These systems can be configured to detect the unique pronunciation, intonation, and other aspects of each speaker's speech. They are often used in situations where a person needs to be identified or authenticated by their speech pattern, such as in biometric identification systems or automatic voice authentication systems. In addition, speaker-oriented systems can be used in speech analysis to study the peculiarities of speaker style or to create personalized voice assistant interfaces that respond to commands or requests of a specific user. Among the common systems, we can highlight: Voice Biometrics by Verint (speech recognition system specializes in identifying a person by his voice, it can identify and authenticate the user based on his unique voice characteristics); Speaker Recognition by NICE (the system uses voice biometric data to identify a person and allows to recognize announcers based on their voice and emphasizes the identification of specific individuals); VoicePIN by Nuance Communications (speech recognition system offers individual voice recognition for user authentication, allows setting a unique ‘voice PIN’ for each user and works with announcers regardless of their speech); VoiceKey by VoiceVault (the system is used for voice authentication of users, it allows to recognize the user's voice even if he uses different phrases or speech patterns).

Speaker-independent speech recognition systems are designed to recognize speech without reference to specific speakers or individuals. They are designed to recognize general speech features and patterns that can be applied to many speakers. These systems are usually trained on substantial amounts of diverse speech data to become more versatile and accurate in speech recognition. Speaker-independent speech recognition systems are widely used in large companies where a large stream of voice commands or data needs to be processed without the need to train a model for each individual user. They can also be used in various applications such as voice assistants, automatic speech recognition systems in the medical or legal fields, as well as in video games and other user interaction scenarios. Among the common systems, we can highlight: Google Speech Recognition (Google offers a widely used speech recognition system that works based on neural networks. It is speaker-independent and capable of recognizing speech from different speakers in different language contexts); Amazon Alexa Voice Service (Amazon's Alexa voice control system is also speaker-independent, and is able to recognize the speech of users from different language areas and with different accents); Microsoft Azure Speech Recognition (Microsoft's Azure speech recognition service offers a scalable and accurate speech recognition system that can work with the speech of different speakers).

1.2. The main tasks of the research and their significance

The purpose of this study is to create a system that can be used to align the Ukrainian text to its audio reproduction. The project will serve as a part of creating a system of Ukrainian language synthesis and recognition of Ukrainian speech. To achieve the goal, the following tasks must be solved: analyze the existing approaches, methods and software tools used in the field of Ukrainian language recognition; identify the main tasks that arise in this case; analyze the methods and algorithms of sign language recognition that can be adapted during system development; implement a system prototype.

The results of the study solve the actual scientific and practical task of harmonizing the text and sound components of Ukrainian-language content, which consists in providing means for effective recognition and reproduction of words in Ukrainian-language fusion speech. Such a system would be useful for a wide range of applications, including speech recognition in audio and video content, development of voice assistants and interfaces, and support for users with disabilities. Given the rapid pace of development of deep learning and natural language processing technologies, the development of such a system has exciting potential for improving the ways of interacting with Ukrainian-language content, ensuring more accurate and faster recognition of Ukrainian speech.

2. Major research results

2.1. Sounds in the Ukrainian language

In the process of research, the mechanisms of phonetics will be used. The main object of the study of phonetics is sounds - the smallest units of the speech stream, which make up words in the language. Sounds form the outer, sound shell of words and thus help to distinguish one word from another. Words are divided by the number of sounds from which they are made, the set of these sounds and their sequence. The sound system of the Ukrainian language includes 38 sounds: 6 vowels and 32 consonants. Speech sounds are produced by the speech apparatus, which includes the larynx with vocal cords, oral and nasal cavities, lips, tongue, teeth, and palate [37].

According to the method of creation, sounds are divided into vowels and consonants [38]. Vowels are the sounds of human speech, the basis of which is the voice. Consonant sounds are the sounds of human speech, the basis of which is noise with a greater or lesser part of the voice or only noise. Active speech organs make certain movements when creating sounds. These are the vocal cords, back wall of the pharynx, uvula (palatal veil), tongue and lips. Active speech organs play the key role in the process of sound formation. Passive speech organs are motionless speech organs approached by active speech organs or even close to them, causing noises. These include the hard palate, teeth, and alveoli. Passive speech organs perform an auxiliary role during sound production [39].

There are six vowel sounds in the Ukrainian language: [і], [и], [е], [у], [о], [а]. They can be:

- Front and back rows. According to the place of production (meaning the movement of the tongue in the horizontal plane of the oral cavity), vowel sounds are divided into front row vowels and back row vowels: front row vowels: [е], [и], [і]; back row vowels: [а], [о], [у].
- Low, medium, and high lift. Depending on the degree of raising the tongue, i.e., on its movement in the vertical plane, vowels of low, medium, and high elevation are distinguished: vowels of low elevation: [а]; middle raised vowels: [е], [о]; high rising vowels: [і], [и], [у].
- Rounded or neutral. With the participation of the lips, vowels are divided into rounded (labialized) and neutral: rounded vowels: [о], [у]; neutral vowels: [і], [и], [е], [а].
- Unstressed and stressed. Depending on the place of stress in the word, vowel sounds can be stressed or unstressed.

Consonant sounds. There are 32 consonant sounds in the Ukrainian language: [б], [п], [д], [д’], [т], [т’], [г], [к], [ф], [ж], [з], [з'], [ш], [с], [с'], [г], [х], [дж], [дз], [дз'], [ч], [ц], [ц'], [в], [й], [м], [н], [н'], [л], [л'], [р], [р']. The division of consonants into loud and sonorous, voiced, and voiceless is based on the participation of voice and noise in their creation. Consonant sounds can be [38]:

1. Voiced and voiceless. Voiceless consonants are consonants in which the voice prevails over the noise. There are nine of these sounds in the Ukrainian language: [в], [й], [м], [н], [н'], [л], [л'], [р], [р']. Voiced consonants, depending on the state of the vocal cords at the time of creation, are divided into voiced and voiceless. If the vocal cords are more tense, then voiced noisy consonant sounds are created. When the vocal cords are relaxed, voiceless sounds are made. voiced consonants: [б], [д], [д'], [ґ], [ж], [з], [з'], [г], [дж], [дз], [дз']; voiceless consonants: [п], [т], [т'], [к], [ш], [с], [с'], [х], [ч], [ц], [ц'].
2. Bilabial, alveolar and glottal. According to the active speech organ, consonants are divided into plosive, alveolar and glottal: bilabial consonants:[б], [п], [в], [м], [ф]; alveolar consonants: [д], [д'], [т], [т'], [з], [з'], [с], [с'], [дж], [дж], [ц], [ц'], [р], [р'], [л], [л'], [н], [н'], [ж], [ч], [ш], [дж], [й];glottal consonant: [г].
3. Hard and soft. According to the sign of hardness or softness, consonants are divided into hard and soft: hard consonants: [б], [п], [д], [т], [ґ], [к], [ф], [ж], [ш], [з], [с], [г], [х], [дж], [ч], [дз], [ц], [в], [м], [н], [л], [р]; soft consonants: [д'], [т'], [з'], [с'], [дз'], [ц'], [й], [л'], [н'], [р']. Separate consonants form pairs according to the 'softness–hardness' feature: hard–soft consonants: [д]– [д'], [т]– [т'], [з]–[з'], [с]–[с'], [дз]–[дз'], [ц]–[ц'], [н]–[н'], [л]–[л'], [р]–[р']. Softened versions of hard consonants, as a rule, appear before a vowel and, however, in a few Ukrainian words and mostly in words of foreign origin, they occur before other vowels.
4. Sibilant, affricate, and nasal consonants. Considering auditory perception, consonants are also divided into sibilant and affricate. A small group consists of nasal consonants, in which the nasal cavity participates. They are divided into sibilant consonants[з], [з'], [с], [с'], [ц], [ц'], [дз] [дз']; affricate consonants: [ж], [дж], [ч], [ш]; nasal consonants: [м], [н], [н'].

2.2. Spectral analysis of an audio fragment

Spectral analysis is one of the signal processing methods that allows characterizing the frequency composition of the measured signal. The Fourier transform is a mathematical basis that connects a temporal or spatial signal (or some model of this signal) with its presentation in the frequency domain. Real-time signal processing includes the tasks of analyzing audio, speech, and multimedia signals, in which, in addition to the difficulties directly related to the analysis of the spectral content and subsequent classification of the sequence of counts (as in the task of speech recognition) or changes in the shape of the spectrum, filtering in the frequency area (mainly refers to multimedia signals), the problem of data flow management in modern computer systems arises. When processing signals, it is customary to solve two types of tasks - detection tasks and evaluation tasks. When detecting, it is necessary to answer the question, whether we are observing a signal with a priori known parameters. Evaluation is the task of measuring the values of the parameters describing the signal [40].

The signal often contains a lot of noise, and interfering signals can be superimposed on it. Therefore, to simplify these tasks, the signal is usually decomposed into the basic components of the signal space. For many applications, periodic signals are of greatest interest. It is quite natural that the functions sin and cos are used. Such decomposition can be performed using the classical Fourier transform [27]. When processing signals of finite duration, interdependent issues must be considered during harmonic analysis. Completion of the observation interval affects the search for tones in the presence of loud noises, the ability to resolve tones of variable frequency, and the accuracy of parameter estimates of all the above-mentioned signals.

Currently, there are many algorithms and groups of algorithms that solve the main task of spectral analysis in one way or another: estimating the power spectral density to judge the nature of the processed signal based on the result. However, each of the algorithms has its own scope of application. For example, gradient adaptive autoregressive methods cannot be applied to data processing with a rapidly changing time spectrum. Classical methods have a wide scope of application but lose to autoregressive methods based on eigenvalues in terms of evaluation quality. However, on a real time scale, the use of the latter is difficult due to computational complexity. Moreover, the application of each of the methods usually requires the selection of parameter values (selection of the data window and correlation window in classical methods, the order of the model in the autoregressive algorithm, the estimated number of eigenvectors in the noise space) and the correct choice requires conducting experiments with each class of algorithms [30].

Thus, the following task arises from existing algorithms, analyze the possibility of application to sequential processing of signals in real time and to block processing and evaluate the obtained results' quality. The statement of the task implies the need to conduct numerous experiments. Experimental input data are formed in the following way: for the task of analyzing the block processing algorithms of the entire sequence of reports, discretized reports of the test-signal data are formed from the sum of complex sinusoids and additive noise processes, formed by passing white noise through a filter with a frequency characteristic of the raised cosine type or a Hamming window. The initial data of the experiments are for the task of analyzing block processing algorithms of the entire sequence. For real-time signal analysis, it is advisable to use power spectral density. The spectral estimate obtained from a finite data record characterizes some assumption about the spectral function that would be obtained if we had a data record of infinite length at our disposal, while accepted statistical criteria for the quality of the estimate are its shift and dispersion.

2.3. Synchronization of the audio stream and phonemic units

Synchronization of the audio stream and phoneme units using the GMM (Gaussian mixture models) algorithm is used in speech recognition tasks. The basic idea is to build a model of the audio stream that can be compared with vectors of phonemic features to determine the correspondence between them. The GMM algorithm uses statistical methods to model the distribution of data in the feature space. In the context of audio stream synchronization and phoneme units, GMM can be used to model various acoustic characteristics of phonemes, such as frequency, amplitude, spectral shape, etc. When the GMM algorithm is trained on a large set of audio data, it becomes able to determine the probability of each phoneme for each part of the audio stream. With the help of these probabilities, it is possible to determine the moments of time when phonemes appear in the input audio stream [41].

The basic idea behind the GMM algorithm discussed in this context is to assume that we know the parameters of this model, and then calculate the probability that each data point belongs to one or another component. After that, the components should be re-aligned so that each component is aligned with the entire data set, each point of which is assigned a weight corresponding to the probability that it belongs to the given component. This process continues iteratively until convergence is reached. The data is ‘supplemented’ by calculating probability distributions for hidden variables based on the current model. When using a mixed Gaussian distribution, the model of the mixed distribution is initialized with arbitrary values of the parameters, and then iterations are carried out according to the two steps described below [42].

1. E-step. Calculate the probabilities $p_{ij} = P(C = i|x_j)$, that data x_j were formed by component i. According to the Bayes rule, the relation $p_{ij} = \alpha P(x_i|C = i)P(C = i)$. The term $P(x_j|C = i)$ represents the probability of data values x_j in the i-th Gaussian distribution, and the term P {C = i) - is a parameter for determining the weight of the i-th Gaussian distribution; by definition $p_i = \sum p_{ij}$.
2. M-step. Calculate the new values of the mathematical expectation, covariance and weight of the component as follows:

$$m_i \leftarrow \sum\nolimits_j \frac{p_{ij}x_j}{p_i}$$

$$s_i \leftarrow \sum\nolimits_j \frac{p_{ij}x_j x_j}{p_i}$$

$$w_i \leftarrow p_i$$

The E-step, or the expectation step, can be considered as the calculation of expected values and hidden indicator variables, where the value is equal to 1 if the data was not formed by the i-th component, and 0 - otherwise. At the M-step, or the maximization step, a search is made for new parameter values that maximize the logarithmic likelihood of the data, considering the expected values of the hidden indicator variables.

The final model, the parameters of which are determined in learning using the GMM algorithm, does not differ from the primary model, on which the data was generated. The logarithmic likelihood of the model obtained in the training process is slightly higher than the corresponding value for the initial model, on which the initial data were formed. This phenomenon may seem strange at first, but it simply reflects the fact that the data was generated randomly, so there is a possibility that it is not an accurate representation of the underlying model. Therefore, the synchronization of the audio stream and phoneme units using the GMM algorithm allows assigning each fragment of the audio stream to the corresponding phoneme, which is a key step in the speech recognition process.

2.4. Mathematical description of the process

The sequence of actions for matching the text and audio components of the Ukrainian-language content was carried out using the algebra of algorithms [43]. The first stage of the implementation of the algebra of algorithms is the description of unit terms and the synthesis of sequences, which is given below.

Formed uniterms: *I(t)* – uniterm of entering/editing text content; *A(t)* - is the uniterm of analysis of the text for the correctness of the specified characteristics; *C(tr)* - uniterm of creating a transcription; *L(a)* - is the uniterm of load/reading of audio content; *As(a)* - uniterm of spectral analysis of the audio track; *F(f)* - uniterm of search for phoneme characteristics; *S(a)* – uniterm of synchronization of transcription and audio track; *V(r)* – uniterm of displaying the result; u_1– check if a value has been entered for analysis; u_2 – checking the correctness of the result. As a result of the use of the apparatus of the algebra of algorithms, the following sequences and eliminations were synthesized:

S_{11} – the sequence of operation of the system in case of availability of values for analysis and a correct result:

$$S_{11}=\overgroup{\overgroup{*\ ,\ \overgroup{C(tr)\ ,\ L(a)}}\ ,\ \overgroup{*\ ,\ V(r)}}$$

S_{12} – the sequence of operation of the system in case of availability of values for analysis and incorrect result:

$$S_{12}=\overgroup{\overgroup{*\ ,\ \overgroup{C(tr)\ ,\ L(a)}}\ ,\ \overgroup{\overgroup{\overgroup{As(a)\ ,\ F(f)}\ ,\ S(a)}\ ,\ V(r)}}$$

S_{21} – the sequence of system operation in the absence of values for analysis and correct result:

$$S_{21}=\overgroup{\overgroup{\overgroup{I(t)\ ,\ A(t)}\ ,\ \overgroup{C(tr)\ ,\ L(a)}}\ ,\ \overgroup{*\ ,\ V(r)}}$$

S_{22} – the sequence of operation of the system in case of no values for analysis and an incorrect result:

$$S_{22}=\left(\begin{array}{l}\overgroup{\overgroup{I(t)\ ,\ A(t)}\ ,\ \overgroup{C(tr)\ ,\ L(a)}}\\ ,\\ \overgroup{\overgroup{\overgroup{As(a)\ ,\ F(f)}\ ,\ S(a)}\ ,\ V(r)}\end{array}\right.$$

L_1 – elimination of check if a value has been entered for analysis:

$$L_1=\overline{S_{11}\ ;\ S_{12}\ ;u_{\bar{1}}\ ?}$$

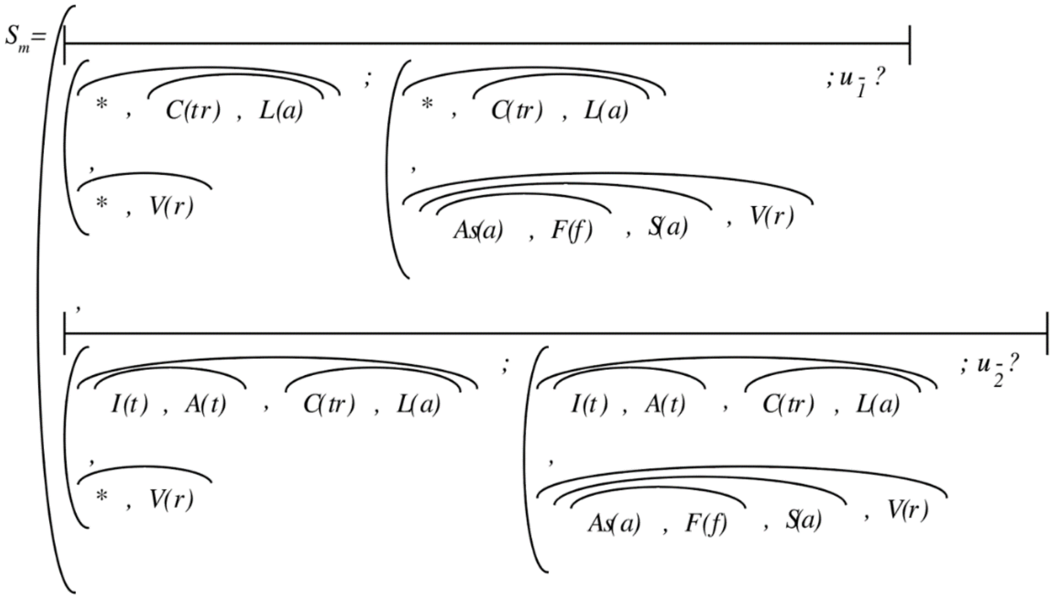
L_2 – elimination of checking the correctness of the result:

$$L_2=\overline{S_{21}\ ;\ S_{22}\ ;u_{\bar{2}}\ ?}$$

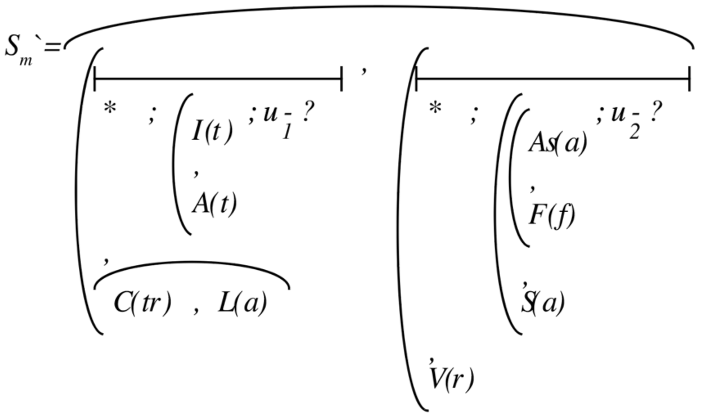
S_m – the main sequence of the system:

$$S_m=\overgroup{L_1\ ,\ L_2}$$

The next stage is the substitution of the corresponding sequences in the elimination.


S_m=
* , C(tr) , L(a)
* , V(r)
;
* , C(tr) , L(a)
As(a) , F(f) , S(a) , V(r)
; u_1 ?
I(t) , A(t) , C(tr) , L(a)
* , V(r)
;
I(t) , A(t) , C(tr) , L(a)
As(a) , F(f) , S(a) , V(r)
; u_2 ?

As a result of using the properties of the algebra of algorithms [14], we subtract the common unit terms by the sign of the elimination operation and obtain the following formula of the algebra of algorithms:



Characteristics of the solution and practical implementation.

The C++ programming language was used to implement the prototype of the software product. It is characterized by such features as: simplicity, object orientation and cross-platform. The main advantages are [44]:

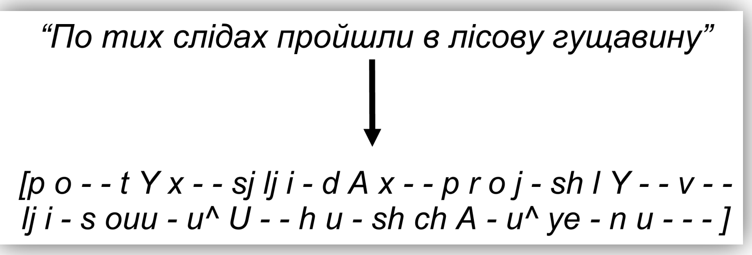
- Scalability. Programs are developed in the C++ language for various platforms and systems.
- Ability to work at a low level with memory, addresses, ports.
- Ability to create generalized algorithms for diverse types of data, their specialization, and calculations at the compilation stage, using templates.
- Various programming styles and technologies are supported, including traditional directive programming, OOP, generalized programming, metaprogramming (templates, macros).

The developed system is presented as a desktop application. The application is created in the environment of the Windows operating system. To carry out this work, the project was divided into two parts: work with text and work with sound.

Working with the text included the following tasks: reading words, applying the rules of assimilation to them, creating a basic transcription, and considering the effects of sounds on each other. Work with audio included: splitting the wave into frequencies, searching for sound parameters, and synchronizing transcription with audio playback. When performing the last task, the GMM (Gaussian Mixture Model) algorithm was used, which helped to achieve high quality results [41].

During prototype testing, different texts were used and read by different voices. The system was configured for fast learning and adapted to different timbres of voices. The requirement for audio reading is the absence of noise and a moderate pace. We will illustrate the operation of the system and display the results of three main stages: creating a transcription, searching for sound characteristics throughout the audio track, and synchronizing text and audio.

As a control example, a fragment of the text was used: *'По тих слідах пройшли в лісну гущавину'* (Following those tracks, they went into the forest thicket). First, let us transcribe the fragment (Fig.1).

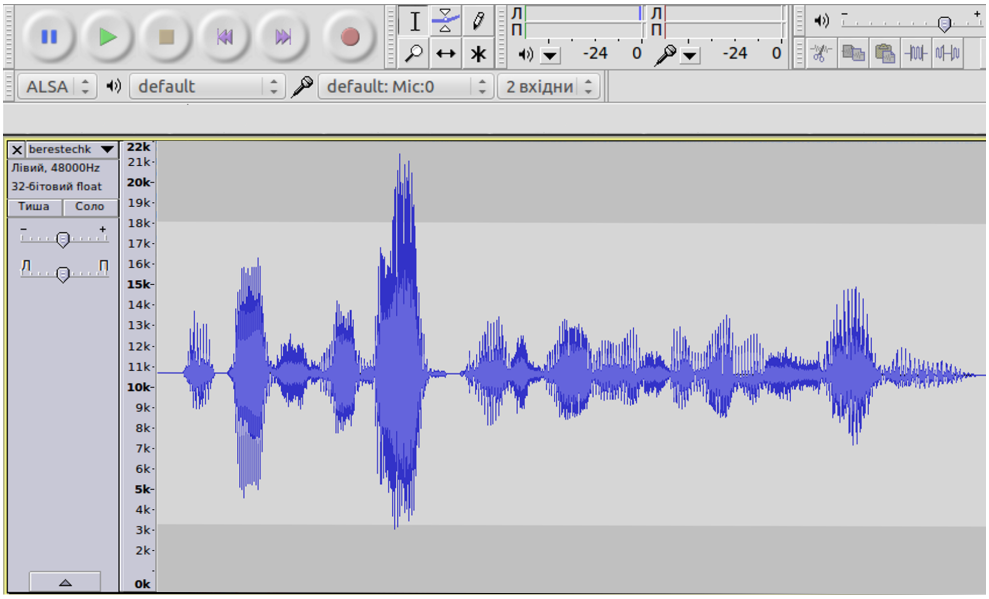


“По тих слідах пройшли в лісову гущавину”

*[p o - - t Y x - - sj lj i - d A x - - p r o j - sh I Y - - v - -
lj i - s ouu - u^ U - - h u - sh ch A - u^ ye - n u - - -]*

Figure 1: Creating a transcription

Next, the audio reading of this fragment is carried out (Fig. 2).



ALSA default default: Mic:0 2 вхідни

berestechk

Лівий, 48000Hz

32-бітовий float

Figure 2: Waveform of the input audio file

After creating the transcription, the wave frequencies are calculated. In Fig. 3, it is noticeable that vowel sounds have a large amplitude of low frequencies, and consonant sounds, such as [t], have a moderately high frequency distribution up to 20 kHz.

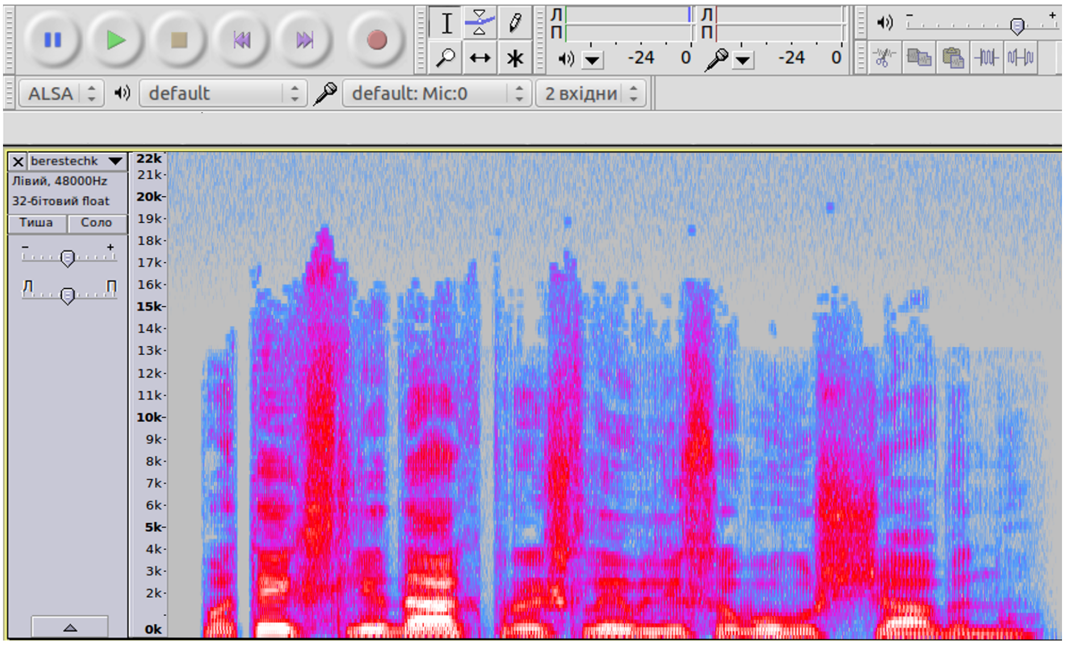


Figure 3: Calculated wave frequencies of the input audio file

Using the GMM (Gaussian Mixture Model) algorithm and predefined phonetic unit characteristics, the text is synchronized with the incoming audio stream. Figure 4 shows the graphical results:

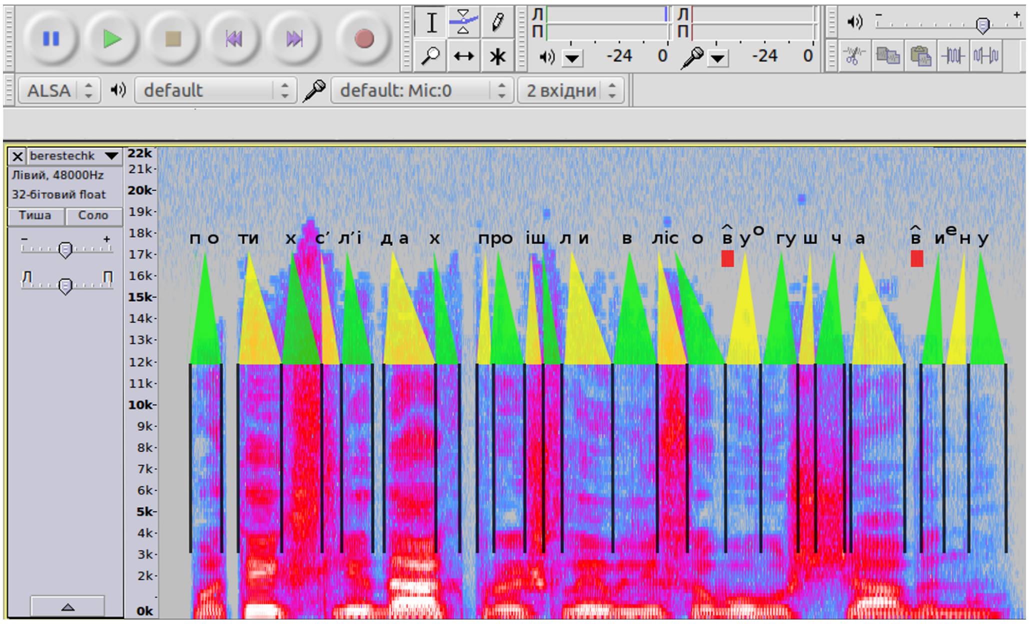


Figure 4: The result of text and audio track synchronization

As it can be noted that the created prototype of the software system successfully compared the text and sound components of the fragment: *'По тих слідах пройшли в лісову гущавину'*.

Conclusion

As a result of the conducted research, existing methods and known systems that provide tools for recognizing the Ukrainian language and describe approaches and methods for synchronizing text and audio information have been analyzed. The stages and software tools for automatic speech recognition were analyzed, which made it possible to identify the features of existing approaches. As the analysis showed, today there are several software systems, but all of them are characterized by certain shortcomings, the main ones of which are limited accuracy in complex language constructions, neglect of the context, and the impossibility of application for the recognition of Ukrainian-language audio content, which makes the task of constructing a system from matching text and audio components of Ukrainian-language content. To present the main aspects of the studied subject area, the classification of sounds in the Ukrainian language was considered, and the features of their detection and formation were given. The next stage was determination of the spectral analysis study and its influence on the recognition process. The stage of the audio stream synchronization and phoneme units using the GMM algorithm is described. The main idea was to build a model of the audio stream that can be compared with vectors of phonemic features to determine the correspondence between them. The mathematical description of the specified process is performed using algebra of algorithms. An applied software system has been developed that implements text and audio information synchronization. At the current moment, the software solution works in the form of a prototype.

Further research will be directed to testing and improving the system, eliminating conflicts, and expanding functionality in accordance with the specified requirements.

References

[1] J. Liao, S. Eskimez, L. Lu, Y. Shi, M. Gong, L. Shou, H. Qu, M. Zeng, Improving Readability for Automatic Speech Recognition Transcription. ACM Transactions on Asian and Low-Resource Language Information Processing. 2023, Volume 22, Issue 5, Article No.: 142., pp 1–23.

[2] M. Danilevsky, S. Dhanorkar, Y. Li, L. Popa, K. Qian, A. Xu, Explainability for Natural Language Processing. KDD '21: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, August 2021, pp. 4033–4034.

[3] L. Minsky, S. Westwater, S. Westwater, Voice Marketing. Rowman & Littlefield Publishers. 2023. P.216.

[4] T. Basyuk, Innerlinking website pages and weight of links. Proceedings of the 12th International Scientific and Technical Conference «Computer Science and Information Technologies CSIT-2017». Lviv, Ukraine, September 12–15, 2017, pp. 12–15.

[5] C. Fan, J. Yi, J. Tao, Z. Tian, B. Liu, Z. Wen, Gated Recurrent Fusion With Joint Training Framework for Robust End-to-End Speech Recognition. IEEE/ACM Transactions on Audio, Speech and Language Processing. 2020, Volume 29, pp 198–209.

[6] J. Dong, Natural Language Processing Pretraining Language Model for Computer Intelligent Recognition Technology. ACM Transactions on Asian and Low-Resource Language Information Processing. 2023, pp. 937–943.

[7] T. Basyuk, A. Vasyliuk, Peculiarities of an Information System Development for Studying Ukrainian Language and Carrying out an Emotional and Content Analysis // CEUR Workshop Proceedings. – 2023. – Vol. 3396: Computational Linguistics and Intelligent Systems 2023: Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems. Volume II: Computational Linguistics Workshop, Kharkiv, Ukraine, April 20-21, 2023. pp. 279–294.

[8] H. Edwards, A. Gregg, Applied Phonetics Workbook: A Systematic Approach to Phonetic Transcription. Cengage Learning. 2003. P.288.

[9] E. Norex, Mastering Dynamic Programming in Python. Independent Creating Platform. 2024. P.219.

[10] J. Keshet, S. Bengio, Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods. Wiley. 2009. P.268.

[11] D. Yu, L. Deng, Automatic Speech Recognition: A Deep Learning Approach (Signals and Communication Technology). Springer. 2015. P.347.

[12] H. Beigi, Fundamentals of Speaker Recognition. Springer. 2011. P.1003.

[13] C. Tran, K. Nguyen-Trong, C. Pham, D. Tran-Anh, T. Nguyen, Improving text recognition by combining visual and linguistic features of text. SoICT '22: Proceedings of the 11th International Symposium on Information and Communication Technology. 2022, pp. 329–335.

[14] P. Kulkarni, Applying Phonetics: Speech Science in Everyday Life. Society Publishing. 2021. P.272.

[15] H. Reetz, A. Jongman, Phonetics: Transcription, Production, Acoustics, and Perception (Blackwell Textbooks in Linguistics). Wiley-Blackwell. 2020. P.400.

[16] J. Dong, Natural Language Processing Pretraining Language Model for Computer Intelligent Recognition Technology. ACM Transactions on Asian and Low-Resource Language Information Processing. 2023, pp.56-78.

[17] L. Tunstall, L. Werra, T. Wolf. Natural Language Processing with Transformers, Revised Edition. O'Reilly Media. 2022. P.406.

[18] T. Basyuk, A. Vasyliuk, V. Lytvyn, O. Vlasenko, Features of designing and implementing an information system for studying and determining the level of foreign language proficiency// CEUR Workshop Proceedings. – 2023. – Vol. 3312: Modern Machine Learning Technologies and Data Science Workshop (MoMLeT&DS 2022): Proceedings of the Modern Machine Learning Technologies and Data Science Workshop, Leiden, The Netherlands, November 25-26, 2022. pp. 212-225.

[19] N. Andreichuk, O. Babeliuk, Contrastive lexicology of English and Ukrainian languages: theory and practice: Textbook. Kherson: Publishing House "Helvetica", 2019. P.236.

[20] U. Kamath, J. Liu, J. Whitaker, Deep Learning for NLP and Speech Recognition. Springer; 1st ed. 2020. P. 649 p

[21] M. Ekman, Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using TensorFlow. Addison-Wesley Professional; 1st edition. 2021. P. 752.

[22] A. Butnaru, Machine learning applied in natural language processing. ACM SIGIR Forum. 2021. Volume 54. Issue 1. Article No.:15. pp 1–9.

[23] B. McFee, Digital Signals Theory. Chapman and Hall/CRC; 1st edition, 2023. P.259.

[24] P. Diniz, Signal Processing and Machine Learning Theory (Academic Press Library in Signal Processing). Academic Press; 1st edition, 2023. P. 1234.

[25] T. Holton, Digital Signal Processing: Principles and Applications Illustrated Edition. Cambridge University Press, 2021. P.1058.

[26] J. Stone, The Fourier Transform: A Tutorial Introduction. Sebtel Press, 2021. P. 103.

[27] R. Goodman, Discrete Fourier And Wavelet Transforms: An Introduction Through Linear Algebra With Applications To Signal Processing. World Scientific Publishing Company, 2016. P. 300.

[28] A. O'Cinneide, D. Dorran, M. Gainza, Linear Prediction: The Problem, its Solution and Application to Speech. DIT Internal Technical Report. 2008. P.19.

[29] P. Alku, R. Saeidi, Rahim, The Linear Predictive Modeling of Speech From Higher-Lag Autocorrelation Coefficients Applied to Noise-Robust Speaker Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2017. pp.1-10.

[30] R. Lyons, Understanding Digital Signal Processing. Pearson; 3rd edition, 2010. P. 954.

[31] L. Tan, J. Jiang, Digital Signal Processing: Fundamentals and Applications. Academic Press; 3rd edition, 2018. P. 920.

[32] P. Manolakis, Digital Signal Processing, Pearson; 4th edition, 2007. P. 1004.

[33] H. Laurent, J. Staines, Taylor Series, Partial Fractions, Laurent Series, and Residues. Independent Creating Platform, 2020. P.44.

[34] C. Pearl, Designing Voice User Interfaces: Principles of Conversational Experiences. O'Reilly Media; 1st edition, 2017. P. 275.

[35] S. Thorn, S. Wei, Instruments of Articulation: Signal Processing in Live Performance. MOCO '19: Proceedings of the 6th International Conference on Movement and Computing, October 2019, pp. 1–8.

[36] A. Vasyliuk, T. Basyuk, V. Lytvyn, Design and Implementation of a Ukrainian-Language Educational Platform for Learning Programming Languages// CEUR Workshop Proceedings. – 2023. – Vol. 3426: Modern Machine Learning Technologies and Data Science Workshop (MoMLeT&DS 2023): Proceedings of the Modern Machine Learning Technologies and Data Science Workshop, Lviv, Ukraine, June 3, 2023. pp. 406–420.

[37] R.-A. Knight, Phonetics: A Coursebook Illustrated Edition. Cambridge University Press, 2012. P.314.

[38] B. Gick, I. Wilson, D. Derrick, Articulatory Phonetics. Wiley-Blackwell; 1st edition, 2013. P.272.

[39] I. MacKay, Phonetics and Speech Science. Cambridge University Press, 2023. P.458.

[40] M. Einsiedler, T. Ward, Functional Analysis, Spectral Theory, and Applications. Springer; Softcover reprint of the original 1st ed., 2018. P.628.

[41] P. Li, C.-H. Zhang, Theory of the GMM Kernel. WW '17: Proceedings of the 26th International Conference on World Wide Web, April 2017, pp. 1053–1062.

[42] K. Kallas, F. Niksic, C. Stanford, R. Alur, Stream processing with dependency-guided synchronization. PPoPP '22: Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, April 2022, pp. 1–16.

[43] V. Ovsyak, Algorithms: methods of construction, optimization, probability research. L'viv: Svit, 2001. P. 268. (In Ukrainian).

[44] B. Forouzan, R. Gilberg, C++ Programming: An Object-Oriented Approach. McGraw Hill; 1st edition, 2019. P. 960.