KAUNAS UNIVERSITY OF TECHNOLOGY

GINTARĖ ŽEKIENĖ

HYBRID RECOGNITION TECHNOLOGY FOR LITHUANIAN VOICE COMMANDS

Doctoral dissertation
Technological sciences, Informatics engineering (T 007)

2021, Kaunas

This doctoral dissertation was prepared at Kaunas University of Technology, Faculty of Electrical and electronics engineering, Department of Automation during the period of 2012–2017 and 2020-2021.

The dissertation defended externally

Scientific Supervisor:
Assoc. Prof. Dr. Kastytis RATKEVIČIUS (Kaunas University of Technology, Technological sciences, Informatics engineering – T 007)

Editor:
Gillian Redfern, UAB "In Public" (English language)
Inga Nanartonytė (Lithuanian language)

Dissertation Defence Board of Informatics engineering (T 007) Science Field:
Prof. Dr. Robertas DAMAŠEVIČIUS (Kaunas University of Technology, Technological sciences, Informatics engineering – T 007) – **chairman**,
Prof. Dr. Arnas KAČENIAUSKAS (Vilnius Gediminas Technical University, Technological sciences, Informatics engineering – T 007),
Prof. Dr. Jurgita KAPOČIŪTĖ-DZIKIENĖ (Vytautas Magnus University, Natural sciences, Informatics – N 009),
Prof. Dr. Olga KURASOVA (Vilnius University, Technological sciences, Informatics engineering – T 007),
Assoc. Prof. Dr. Tomas KULVIČIUS (University of Goettingen, Germany, Natural sciences, Informatics – N 009).

The official defence of the dissertation will be held at 1 p.m. on April 22, 2021 at the public meeting of Dissertation Defence Board of Informatics Engineering Science Field in Dissertation Defence Hall at Kaunas University of Technology.

Address: K. Donelaičio St. 73-403, 44249 Kaunas, Lithuania.
Tel. no. (+370) 37 300 042; fax. (+370) 37 324 144; e-mail doktorantura@ktu.lt.

Doctoral dissertation was sent on March 22, 2021.
The doctoral dissertation is available at the libraries of Kaunas University of Technology (K. Donelaičio St. 20, Kaunas, Lithuania), Vilnius Gediminas Technical University (Saulėtekio al.14, Vilnius) and internet (http://ktu.edu).

KAUNO TECHNOLOGIJOS UNIVERSITETAS

GINTARĖ ŽEKIENĖ

HIBRIDINĖ LIETUVIŠKŲ BALSO KOMANDŲ ATPAŽINIMO TECHNOLOGIJA

Daktaro disertacija
Technologijos mokslai, informatikos inžinerija (T 007)

2021, Kaunas

Disertacija rengta 2012–2017 ir 2020-2021 metais Kauno technologijos universiteto Elektros ir elektronikos fakulteto Automatikos katedroje.

Disertacija ginama eksternu

Mokslinis konsultantas:
Doc. dr. Kastytis RATKEVIČIUS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – T 007)

Redagavo:
Gillian Redfern, UAB "In Public" (anglų kalbos redaktorė)
Inga Nanartonytė (lietuvių kalbos redaktorė)

Informatikos inžinerijos mokslo krypties disertacijos gynimo taryba:
Prof. dr. Robertas DAMAŠEVIČIUS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007) – **pirmininkas**,
Prof. dr. Arnas KAČENIAUSKAS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – T 007),
Prof. dr. Jurgita KAPOČIŪTĖ-DZIKIENĖ (Vytauto Didžiojo universitetas, gamtos mokslai, informatika – N 009),
Prof. dr. Olga KURASOVA (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – T 007),
Doc. dr. Tomas KULVIČIUS (Gottingeno universitetas, Vokietija, gamtos mokslai, informatika – N 009).

Disertacija bus ginama viešame informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje 2021 m. balandžio 22 d. 13 val. Kauno technologijos universiteto disertacijų gynimo salėje.

Adresas: K. Donelaičio g. 73-403, 44249 Kaunas, Lietuva.
Tel. (370) 37 300 042; faks. (370) 37 324 144; el. paštas doktorantura@ktu.lt.

Disertacija išsiųsta 2021 m. kovo 22 d.
Su disertacija galima susipažinti Kauno technologijos universiteto (K. Donelaičio g. 20, Kaunas), Vilniaus Gedimino technikos universiteto (Saulėtekio al.14, Vilnius) bibliotekose be internete (http://ktu.edu).

ABBREVIATIONS

ANN	Artificial neural network
ASR	Automatic speech recognition
DNN	Deep neural network
DTW	Dynamic time warping
GMM	Gaussian mixture model
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
ICD-10-CM	International Classification of Diseases (10th revision)
IPA	International Phonetic Alphabet
IVR	Interactive voice response
MFCC	Mel-frequency cepstral coefficients
MSS	Microsoft Speech Server
RA	Recognition accuracy
REC_LTp	HTK phoneme-based Lithuanian recognizer
REC_LTt	HTK triphone-based Lithuanian recognizer
REC_LTtri	INFOBALSAS HTK triphone-based Lithuanian recognizer
REC_LTw	HTK word-based Lithuanian recognizer
REC_MSS	MSS-based recognizer
REC_SP	Windows OS Spanish recognizer
SAMPA	Speech Assessment Methods Phonetic Alphabet
UPS	Universal phone set

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. The importance of the work

It is well known that speech recognition-based interfaces could have great value in many applications.

This is particularly true for applications oriented towards telecommunication users. A speech input interface based on speech recognition, also called a Voice User Interface (VUI), has already been used in many applications. This can be particularly beneficial for use by disabled people, as automatic speech recognition (ASR) is potentially of enormous benefit to people with severe physical disabilities. The tremendous richness of human speech gives the user many degrees of freedom for control and input. The speed of speech recognition also gives it a potential advantage over other input methods commonly employed by physically disabled people (1).

In recent years, speech recognition technologies have been widely applied throughout information technologies. Therefore, the application of speech recognition in information technologies is an extensively explored area. Systems for the recognition of the spoken language are applied in various areas. In the automobile industry, for example, speech recognition is used in hands-free equipment, the management of navigation and multimedia devices, and smart phone connectivity. Successful research in the area of speech recognition requires significant financial resources and a large amount of data covering the complexity of voice commands. Naturally, large companies such as Google successfully conduct studies in the area of speech recognition, which are based on the principle of Hidden Markov Models (HMMs), and produce surprising results even in the recognition of languages that are not widely used. Google's success is determined by a large number of collected speech corpora that are used for the training of recognition systems.

The Lithuanian language is not one of the more widely used languages in the world, and other countries therefore do not prioritize its study nor the application of speech recognition to it. However, this does not stop Lithuanian scientists and researchers from implementing this idea of adapting existing products for speech recognition in their own studies.

1.2. The relevance of the problem

The main problem of the recognition of disease names was defined during the project "Hybrid Recognition Technology for Voice Interface INFOBALSAS" (hereinafter – the INFOBALSAS project), which ended in 2013. The main goal of the project was to develop hybrid voice command recognition technology and implement it in the first practical informative service that used the recognition of Lithuanian voice commands. The Lithuanian medical information system that was developed was able to recognize the names of the most commonly encountered diseases, the most common pharmaceuticals, and the most frequent complaints in the medical practice. The total number of voice commands implemented in the system was approximately 1,000.

The list of diseases approved by the Ministry of Health contains approximately 15,000 diseases and disorders. So far, we have not been able to well recognize such an extensive list of diseases. The solution to the problem of disease recognition is to use codes from the International Classification of Diseases (ICD-10-CM) to recognize codes containing several letters and digits. This technique also allows for the recognition of digit names and letters for application with other types of codes.

1.3. The goal of the work

The goal of this thesis is to create a hybrid recognition technology for Lithuanian voice commands that connects two or more speech recognizers. It is expected that, as a result of connecting different recognizers, if one recognizer makes a mistake then another/others will make the correct decision.

1.4. The tasks of the work

1. To collect Lithuanian digit names and names starting with the 26 letters of the Latin alphabet to form a speech corpus appropriate for identifying codes consisting of digits and Latin letters.
2. To adapt the non-native language recognizer for the recognition of Lithuanian voice commands.
3. To create two Lithuanian recognizers using word-based and phoneme-based HMMs.
4. To connect two or more recognizers using machine learning methods.
5. To compare the results of the accuracy of recognition with the results of similar research carried out in Lithuania.

1.5. The methods and tools of research

The Lithuanian recognizer was modeled with the HTK toolkit of word-based, phoneme-based, and contextual phoneme-based HMMs. The MFCC features of the HMMs were selected to ensure strong results in the recognition of isolated word commands. The freely distributed Windows 7 and Windows 8 (8.0 (Spanish-US)) Spanish language recognizer was selected as the non-native recognizer. The Spanish language recognizer (9.0 for MSS (Spanish-US)) of Microsoft's Speech Server (MSS'2007) was selected for telephone applications. The freely distributed WEKA packet was selected for the connection of recognizers.

The techniques for selection of Lithuanian names were prepared based on the results of an investigation into the Spanish language 8.0 recognizer's recognition of Lithuanian names.

1.6. The scientific novelty of the work

- A selection technique of names and words that is appropriate for the identification of Latin letters by recognizing proposed names or words was created. This technique ensures over 30% increased accuracy in the identification of Latin letters compared to the NATO alphabet.

- The technique of the connection of recognizers was created using machine learning methods. It operates by combining features obtained from recognizers with additional features which depend on the recognized word.

This methodology was tested in the following ways:

- by combining the recognition results of five different speech corpora or their fragments:
  a) two speech corpora containing digit names (30 speakers, 10 digits, 20 pronunciations each; and 50 speakers, 10 digits, 1 pronunciation each);
  b) a speech corpus of names (21 speakers, 26 names or words, 20 pronunciations);
  c) a speech corpus of medical terms (12 speakers, 731 phrases or words, 20 pronunciations each);
  d) a speech corpus of phrases and words (143 speakers, 18 phrases, 8 words, 1 pronunciation each);
- by combining the recognition results of the names speech corpus using different engines (Microsoft and Baidu);
- by combining the recognition results of the names speech corpus with a signal:noise ratio of 5 dB;
- by combining the recognition results of the digit names speech corpus using a telephone format (8 kHz, 8 bits).

Three packages were used in the recognition studies: HTK, Kaldi, and TensorFlow.

1.7. The practical significance of work results

The results of this research could be applied and used in the development of ASR systems for applications involving the recognition of codes.

One of these applications could be the recognition of disease names according to their codes (using ICD-10-CM), consisting of one letter and several digits. Examples of codes containing only digits that could be recognized include PIN, personal identification codes, etc. Major attention is paid to the recognition of digits because the use of digits is dominant in codes, and very high digit recognition accuracy is required. In order to reach a recognition accuracy of approximately 90% in a 10-digit sequence, a recognizer should recognize individual digits with 99% accuracy.

Another potential application is the recognition of codes containing only digits through the telephone.

The proposed method of connecting recognizers was implemented in the new hybrid recognition technology created and demonstrated during the INFOBALSAS project.

1.8. Defensive statements

1. The proposed technique for the selection of names and words is appropriate for the identification of Latin letters. It ensures over 30% increased accuracy in the recognition of the names and words speech corpus (21 speakers, 26 names and

words, 20 utterances) compared to the recognition accuracy of the NATO alphabet speech corpus (2 speakers, 26 words, 50 utterances).

2. The proposed technique for the connection of several recognizers involved using the machine learning method and combining features obtained from the recognizers and additional features which depended on the recognized word. This enabled the improvement of the recognition accuracy of all speech corpora used in the recognition experiments. The main aspects of this proposed technique were:
   - the features, extracted from all speech corpus, were used in the process of classification. This was the main difference compared to the other methods of connecting several recognizers;
   - some additional features (sp_supp, lt_delta_prob, gender, lt_a, …, lt_ž, sp_a, ..., sp_ž) were used. They were produced by speech experts either manually or by using the outputs of recognizers. Such features increased classification accuracy in all cases compared to features produced by the outputs of recognizers alone;
   - the results of research involving the connection of two or three recognizers, using the abovementioned speech corpora, showed that the suggested method improved the accuracy of the hybrid recognizer in all cases;
   - the proposed technique for the connection of several recognizers was tested using the medical speech corpus, consisting of isolated words and phrases. The RIPPER classifier and proposed hybrid recognizer decreased recognition error by 24% compared with the HTK-based Lithuanian recognizer alone.

1.9. The author's participation in the project

The author participated in the 2011–2013 High Technology Development Program in the INFOBALSAS project.

1.10. The approbation of work results

The following articles were published in journals indexed in the Web of Science with Impact Factor:

1. Bartišiūtė, Gintarė; Ratkevičius, Kastytis. Speech server based Lithuanian voice commands recognition // Electronics and electrical engineering. Kaunas: KTU. 2012, Vol. 18, no. 10, p. 53–56.
2. Rudžionis, Vytautas Evaldas; Raškinis, Gailius; Maskeliūnas, Rytis; Rudžionis, Algimantas Aleksandras; Ratkevičius, Kastytis; Bartišiūtė, Gintarė. Web services based hybrid recognizer of Lithuanian voice commands // Electronics and electrical engineering. Kaunas: KTU. 2014, Vol. 20, no. 9, p. 50–53.
3. Bartišiūtė, Gintarė; Paškauskaitė, Gintarė; Ratkevičius, Kastytis. Advanced Recognition of Lithuanian Digit Names Using Hybrid Approach // Electronics and electrical engineering. Kaunas: KTU. 2018. (Accepted for publication)

The following publications appear in other international databases and were presented at 6 scientific conferences in Lithuania and abroad:

1. Bartišiūtė, Gintarė; Ratkevičius, Kastytis. Investigation of Lithuanian digit names recignition accuracy // Electrical and control technologies: proceedings of the 8th international conference on electrical and control technologies ECT 2013, May 2–3, 2013, Kaunas, Lithuania / Kaunas University of Technology, Technologija. 2013, p. 9–12.
2. Rudžionis, Vytautas; Raškinis, Gailius; Ratkevičius, Kastytis; Rudžionis, Algimantas Aleksandras; Bartišiūtė, Gintarė. Medical – pharmaceutical information system with recognition of Lithuanian voice commands // Human language technologies – the Baltic perspective: proceedings of the 6th international conference, Baltic HLT 2014, Kaunas, IOS Press. (Frontiers in artificial intelligence and applications, vol. 268, p. 40–45.
3. Bartišiūtė, Gintarė; Paškauskaitė, Gintarė; Ratkevičius, Kastytis. Investigation of disease codes recognition accuracy // Proceedings of the 9th international conference on Electrical and Control Technologies, ECT 2014 / Kaunas University of Technology, Kaunas: Technologija. 2014, p. 60–63.
4. Bartišiūtė, Gintarė; Ratkevičius, Kastytis; Paškauskaitė, Gintarė. Hybrid recognition technology for isolated voice commands // Information systems architecture and technology: Proceedings of 36th international conference on information systems architecture and technology, ISAT 2015, (Springer, 2016, Advances in intelligent systems and computing, vol. 432, p. 207–216.
5. Bartišiūtė, Gintarė; Paškauskaitė, Gintarė. Šnekos atpažintuvų sujungimo galimybių tyrimas // E2TA-2015: Elektronika, elektra, telekomunikacijos, automatika: 12th student scientific conference on electronics, energy, telecommunications and automation. Kaunas: Kaunas University of Technology, 2015. p. 20–23.
6. Ratkevičius, Kastytis; Paškauskaitė, Gintarė; Bartišiūtė, Gintarė. Recognition of ICD-10 codes by combining two recognizers // Frontiers in artificial intelligence and applications: Human language technologies – the Baltic perspective: proceedings of the seventh international conference Baltic HLT 2016, vol. 289, p. 51–58.

1.12. The structure of the dissertation

This dissertation consists of six sections, a literature list (174 references), and six annexes.

The introduction discusses the importance of the work, the relevance of the problem, the tasks of the work, the methods and tools of research, the scientific novelty, the practical significance of the results obtained, defensive statements, the author's participation in projects, the approbation of the work, the structure of the work, and the contents.

The second section provides an overview of speech recognition technology, involving a discussion of hidden Markov chains, dynamic correction of the time axis, and neural networks. Hybrid technology and classifiers are examined, and the speech corpora are overviewed.

The third section describes the methods for the formation of Lithuanian voice command transcriptions. The algorithms for the selection of names and words, and the methods for the connection of recognizers are provided. The speech corpora used in the research and software are also described.

The fourth section describes recognition studies with different recognizers using the speech corpora of digits and names, and results and conclusions are presented.

The fifth section presents research on the connection of two or three recognizers using the digits, names, and medical speech corpora, and the LIEPA speech corpus.

In the final conclusions section, the summarized results of the dissertation are presented.

2. OVERVIEW AND ANALYSIS OF SPEECH RECOGNITION TECHNOLOGY

For most of human history, speech has been and remains the main instrument for communication between humans, and is widely used for information exchange, negotiation, etc. Research in the processing and recognition of speech, for the most part, has been motivated by people's desire to build machines or mechanical models that can emulate the actions and verbal communication abilities of humans. The key moment in this endeavor was the appearance of the first computers, and with them artificial intelligence. The rapid development of information technologies demanded a natural language-based user interface to accompany the hardware. Today, radios, TVs, telephones, transport systems, computers, and satellite technology all play a key role in our daily lives. They also break down the walls between countries and bring people together via new platforms of communication. This creates access to various sources of information – whether sounds, images, texts, or otherwise – from anywhere in the world. In some countries, the functionality to book tickets by voice, ask for the train schedule via telephone (2), receive tourist information, or use automatic translation services (3) already exists.

However, as yet no voice dialogue systems which are able to talk with anyone about anything exist. The most advanced form of speech technology is still limited to natural spoken-language answers to specific questions. More commonly, speech recognition technology applications are emerging in contact center systems or various mobile devices – for example, in smart phones (4), spectacles (5), hearing aids (6), or cochlear implants (7). For many people with disabilities, speech is perhaps their only source of communication. Websites and programs with expanded voice dialogue opportunities allow these people to surf the internet and manage information on compatible devices via the use of spoken commands (8). IT industry giants such as Microsoft and IBM have already delivered speech applications for their own software platforms – Microsoft Speech Server (MSS) and IBM Websphere Voice Server, respectively. The following overview provides an examination of the development of the most significant speech recognition technologies, both around the world and in Lithuania.

2.1. Recognition problems

Speech recognition requires a lot of time and effort. A variety of methods have been developed and a lot of systems have been realized, some of which have been applied already, yet still a lot of difficulties and unanswered questions remain when realizing precise, noise-resistant detection systems. These difficulties can be characterized by several problems, one of which is the variability of speech signals – i.e., the impossibility of realizing two completely identical examples of the same linguistic unit. Put simply, it is impossible to pronounce the same word in the same way – even if one were to attempt to do so indefinitely, pronunciation will differ in pace, energy level, or any other temporal or spectral characteristics. There are two types of variability of speech: internal and external. Internal variability occurs within

the instability of the speaker's speech, and one of the reasons for this variability is the speaker's manner of speaking. A speaker can express their thoughts with a raised tone, shouting, whispering, trying to conceal their accent, and so on. In addition, speech is influenced by subjective factors such as the speaker's posture, mood, health condition, age, or attitude to the topic of conversation. Due to these reasons, even words pronounced by the same person will differ, and these differences will increase as time passes. The natural features of language (co-articulation, the intervention of meaningless sounds, variations in speech tempo etc.) contribute to the abovementioned reasons for internal variability. Between speakers of different sexes and different ages, acoustic differences are especially prevalent. Speech variability can be solved in two ways: the first way is the adaptation of the speaker; the second way is the use of a system with speaker-resistant features.

The second problem is characterized by the features of natural speech. One phenomenon of natural speech is co-articulation: a fusion of adjacent sounds. Fused sounds become difficult to separate, or even acquire the sound of a completely different phonetic unit (for example, the word "čia" we hear as "če," and only grammar dictates that we write it correctly). Non-linguistic sounds are also characteristic of natural speech (a cough, for example, or a "hmmmm" when in doubt), and can fill pauses, intervene into a word, or even prevent its occurrence entirely. Human perception easily distinguishes these sounds as non-linguistic, while a recognition system can understand them as a word or a part thereof (especially if the results of acoustic analysis present the sound in such a way). In some cases, the absence of boundaries between words may be relevant. These issues should be solved at the linguistic level by using speech models and applying additional knowledge of grammar, semantics, and pragmatics. Therefore, in addition to processing the signals of acoustic speech, the need for linguistic processing occurs. Dictionaries of recognition systems are another source of problems in recognition – large dictionaries are confusing, as they often contain a lot of acoustically similar examples. Some state that the difficulty of the speech recognition task increases logarithmically to the increase in the size of the dictionary (9). One possible solution to this problem is the use of a context (i.e., one that is designed for a specific subject) dictionary.

The problem of words that are absent from a dictionary is even more difficult to solve. Any system will eventually face a word that is absent from a dictionary. In such a case, there are two different solutions: reject the word as unrecognized; or include it into the system's dictionary. The second solution leads to a set of issues that have hardly been resolved: how to guarantee that the unrecognized example is linguistically meaningful; how to generate the required transcription; how to distinguish an example from extraneous noise, etc. As yet, there are no effective procedures that exist to address these issues and, as a result, unrecognized examples are simply ignored.

The fourth problem is the influence of signal acoustics and environmental spread on a signal. Any noise present at the signal generation, spread, and reception stages can and will influence the signal. Sources of noise can include: the speaker themselves (exhalation, the mechanical noises of organs of speech), environmental spread (background noise, echoes), input device (electrical noise of the microphone, nonlinear distortions), transmission channel (reflections, nonlinear distortions of

channel), and receiving device (electrical noise, nonlinear distortions, quantization noise). The result of all of these factors is a noisy signal that, whilst easily understandable to humans, is sometimes totally unacceptable for a technical system. In addition, each device has its own individual spectral characteristics (limited bandwidth, for example) which also influence the processed signal and therefore the algorithm's analysis of the qualitative characteristics of the speech. As such, the impact of the various technical characteristics (different purposes, different manufacturers) of a signal can vary. A system trained with one (perfectly operational) microphone can completely lose its properties if the microphone is replaced (in this case, the efficiency of the system depends on equipment). This problem should be resolved by searching for noise-tolerant systems.

In summary, it could be said that automatic recognition systems are not comparable to human speech perception due to the abovementioned problems, amongst others. A person does not limit themselves in communication with acoustic analysis. Instead, when communicating, they use their knowledge of phonetics, phonology, lexicon, syntax, semantics, pragmatics, as well as the contextual data of the conversation. They also acquire additional information transmitted by gestures, facial expressions, posture, perhaps even their intuition, and other intangible sources of information which cannot be realized with a technique due to ignorance, complexity, or our own prior assumptions.

2.2. Speech recognition technology

Before discussing the future of ASR systems, it is important to ascertain what has already been achieved in this area, which technologies and methods are most common, and to identify the possible practical problems of their application. In this and subsequent sections, the classification of speech recognition systems, the basics of their operation, and an analysis of scientific research conducted by both Lithuanian and international authors in this field are presented.

It should first be highlighted that a speech signal is not stationary. The spectral density of speech changes in time depending on the position of the glottis signals (e.g., by influencing the main tone) and speech organs (tongue, lips, etc.). For example, such a signal can be modeled based on HMMs as a sequence of certain stationary random events. In the first stage of signal processing, most ASRs analyze a short fragment of the signal, according to which the stationary speech is determined. For the analysis of the signal, various filters are widely used alongside a cepstral analysis that looks for specific features. Compensation for the influence of noise may be carried out at several levels, including: processing the speech signal; training models using the speech corpora alongside noisy records; treating the noise as the missing information that can be removed in statistical models; or evaluating parametric distributions between noise and speech. Even in well-articulated speech, the acoustic realization of certain phonemes depends on the constant movement of articulators, which is itself dependent on past and future phonemes. Analyzing the influence of coarticulation and pronunciation, spectral characteristics, and the gender of the speaker are distinguished as yet more essential dimensions of speech change. Figure 2.1 shows how certain factors influence recognition systems.

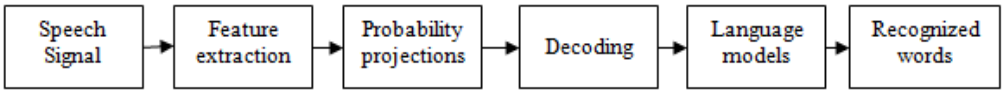


Figure 2.1. The process of speech recognition

The first block is composed of the acoustic environment and transmission equipment (microphone, signal amplifier, filter, etc.). Their quality can significantly affect the subsequent processes. The second block (the subsystem involving the receipt of signs) is projected in order to find acoustic representations and the specific features of the speech signal. This takes place in order to qualitatively distinguish classes of speech sounds and to effectively reject unnecessary variations. The next two blocks illustrate operations for the collation of acoustic features. In almost all ASRs, spectral or cepstral reports of the speech signal (features) are calculated at certain intervals – 100 times per second, for example. In order to recognize speech, the signs obtained are compared with those obtained from the training data using a certain measure of similarity or distance. Each of these comparisons represents a local measure. The global measure is the search for the best sequence of words, which is mostly determined by combining parts of local measures (e.g., an entire word is searched for according to a set of phonemes). The local equivalent usually does not present the one best choice, but instead offers a group that corresponds to possible sounds. Another function of the decoding block is compensation for temporary distortions that occur in normal speech. For example, vowels are usually shortened when speaking rapidly, whilst consonants remain the same length.

Signal and ASR systems are influenced by other linguistic variables. A human can speak quieter or louder, faster or slower, etc. In addition, certain reflex effects can be distinguished, such as speaking louder in a noisy environment.

Speaking faster or slower also influences the speech signal, influencing both the temporal and spectral characteristics of the signal whilst also influencing acoustic models. Naturally, in the speech of a person who is speaking quicker, pronunciation changes occur quicker and more often. Speech also changes depending on age, generation, and for physiological reasons.

Dictionaries for the training of models are not usually formed of recordings from children or elderly people, so recognition errors are expected for these age groups when using ASR systems. Emotions also have a significant impact on the quality of ASR, as their recognition can allow us to identify the emotional state of the user. The abilities of newer systems to recognize spontaneous conversations allow us to distinguish the influence of this style of speaking, and to better characterize the phenomenon of variations in pronunciation expressed in spontaneous speech.

2.2.1. The classification of speech recognition systems

Speech recognition systems can be divided into several different classes according to how many words they can recognize:

1. Recognition systems for voice commands with isolated words. The phrase that is recognized can be formed of more than one word (e.g., a sentence), but, when

speaking, the user is required to leave a brief pause after each pronouncement, similarly to how one might read a telegram (e.g., “open” – STOP – “close”). The systems where the user has to wait until a word is recognized are also called isolated utterance systems.

2. Recognition systems for connected words. Such systems are similar to the recognition systems for isolated word commands; however, the silent spaces between words are much smaller (often used for the recognition of a sequence of numbers).

3. Recognition systems for continuous speech. Such systems allow the user to speak almost naturally, and are most commonly applied for the recognition of dictation.

4. Recognition systems for spontaneous speech, which are capable of processing the properties of natural speech – including multiple words being pronounced as one, the use of words which have no meaning without context, and even stuttering. It is very difficult to recognize such speech, and so most examples of this type of ASR are still in the prototype stage. The goal of any ASR system is to recognize spoken words as accurately as possible, i.e., theoretically no worse than a human does it, regardless of the voice characteristics of the speaker, the size of their vocabulary, data transmission conditions, etc. However, most ASRs reach a recognition accuracy of over 90% only when there are certain conditions met. For example, accuracy in the recognition of some number names using a microphone, a small dictionary, and an environment without any background noise.

The recognition algorithm is usually based on statistical models, and HMMs are common. HMMs are generally defined as stochastic finite state automations, and it is assumed that they are formed of a finite set of possible states, where each state corresponds to a certain distribution of probability (in the case of similarity, the function of probability density). Ideally, a separate HMM should be composed for each word. However, in practice, this is difficult to implement, and so instead a sentence is modeled as a sequence of words. Some ASRs work at the word level, but at the level of a larger dictionary they usually use parts of words, thereby reducing the number of required parameters and the training data. Each word can be divided into a certain group of acoustic units. Usually, it is divided into phonemes, and then into vowels – consonants cases are also possible. One or more HMM states are used in order to model the phoneme corresponding to the speech segment. Word models consist of a chain of phonemic models which are limited by the dictionary, and models of sentences consist of a chain of word models which are limited by the rules of grammar.

First, a discredited speech signal is transformed into a set of features at a fixed ratio, typically every 10–20 ms. According to these parameters, it then usually looks for candidates among the most similar words, introducing acoustic, lexical, and speech-model constraints. In this process, the training data are used to determine the values of model parameters. At the level of signal representation, certain properties of signals independent of the human voice are distinguished, and dependent properties are separated. At the acoustic-phonetic level, speech variability is typically modeled using statistical techniques for large amounts of data. The impact of linguistic context on the acoustic-phonetic level is usually managed by training the individual models

of phonemes in various contexts. This method is also called context-dependent acoustic modeling. Variations at the word level can be managed using the alternative pronunciation of words in a manner similar to pronunciation networks. Typical alternative pronunciations of words (and the effects of dialect and accent) are processed, with search algorithms directed to look for alternative pronunciation in the networks of several phonemes. Statistical speech models that are based on the probabilities of the appearance of certain "word" sequences are used to search through all possible sequences of words.

2.2.2. Review of the evolution of speech recognition technology

In this chapter, the aim is to review the major results of research and the practical applications of science in the field of speech recognition, influenced by today's ASR systems and their prevalence.

The earliest attempts to design systems for ASR were made in the 1950s and 1960s, when various researchers were guided mostly by the theory of acoustic-phonetics, which describes the phonetic elements of speech (the basic sounds of language) and tries to explain how they are acoustically realized in a spoken utterance (10). Since signal processing and computer technologies were still very primitive, most speech recognition systems were investigated using spectral resonances during the vowel region of each utterance, which were extracted from output signals of an analogue filter bank and logic circuits (11).

In 1952, Davis, Biddulph, and Balashek built a system for the isolated digit recognition of a single speaker at Bell Laboratories (12), using the formant frequencies measured and estimated during the vowel regions of each digit. In other early ASR systems of the 1950s, Olson and Belar of RCA Laboratories, USA, tried to recognize ten distinct syllables of a single speaker, as embodied in ten monosyllabic words (13). In 1959, at University College in England, Fry and Denes tried to build a phoneme recognizer to recognize four vowels and nine consonants (14). By incorporating statistical information concerning allowable phoneme sequences in English, they increased the overall phoneme recognition accuracy for words consisting of two or more phonemes. This work marked the first use of statistical syntax (at the phoneme level) in ASR. Another notable effort towards recognition in this period was achieved at the MIT Lincoln Lab, when Forgie and Forgie built a speaker-independent 10-vowel recognizer (15).

In the 1960s, several Japanese laboratories demonstrated their ability to construct purpose-built hardware for performing speech recognition. Most notable were the vowel recognizer of Suzuki and Nakata at the Radio Research Lab in Tokyo (16), the hardware phoneme recognizer of Sakai and Doshita at Kyoto University, which used a hardware speech segmenter and a zero-crossing analysis of different regions of input (17), and the digit recognizer at NEC Laboratories (18).

One problem of speech recognition exists in the nonuniformity of time-scales in speech events. In the 1960s, the first efforts were made by Martin and colleagues at RCA Laboratories (19), and Vintsyuk in the Soviet Union (20). Martin developed a set of elementary time normalization methods, based on the ability to reliably detect the start and end points of speech, that significantly increased recognition

performance (19). Vintsyuk proposed the use of dynamic programming (known as dynamic time warping – DTW) for time alignment between two utterances in order to derive a meaningful assessment of their similarity (20). Vintsyuk also proposed algorithms for connected word recognition. However, his work was largely unknown in other countries until the 1980s. At the same time, Sakoe and Chiba (21) started to use a dynamic programming method in speech pattern matching. Since the late 1970s, mainly due to the publication of Sakoe and Chiba, dynamic programming, in numerous variant forms (including the Viterbi algorithm (22), which came from the communication theory community), has become an indispensable technique in ASR.

In the late 1960s, Atal and Itakura (23, 24) independently formulated the fundamental concepts of Linear Predictive Coding (LPC), which greatly simplified the estimation of the vocal tract response from speech waveforms. By the mid-1970s, the basic ideas of applying fundamental pattern recognition technology to speech recognition, based on LPC methods, were proposed by Itakura while working at Bell laboratories (25), and by Rabiner and Levinson (26) amongst others. Simultaneously, in the late 1960s, Reddy conducted pioneering research at Carnegie Mellon University (CMU) in the field of continuous speech recognition using the dynamic tracking of phonemes (27).

Martin ultimately founded one of the first speech recognition companies, Threshold Technology, which built, marketed, and sold speech recognition products. Their first real ASR product was called the VIP-100 System. The system was only used in a few simple application fields, such as television faceplate manufacturing companies (for quality control) and FedEx (for package sorting), but its main importance was in the way that it influenced the Defense Advanced Research Projects Agency (DARPA) of the U.S. Department of Defense to fund the Speech Understanding Research (SUR) program, along with many seminal systems and technologies (28), during the early 1970s. One of the first demonstrations of speech understanding was achieved by CMU in 1973. Their Harpy system (29) was shown to be able to recognize speech with reasonable accuracy using a vocabulary of 1,011 words. One particular contribution from the Harpy system was the concept of graph search, where the speech recognition language is represented as a connected network derived from lexical representations of words, with syntactical production rules and word boundary rules. In the proposed Harpy system, the input speech, after undergoing a parametric analysis, was segmented, and the segmented parametric sequence of speech was then subjected to phone template matching using the Itakura distance (30). Other systems developed under DARPA's SUR program included CMU's Hearsay II and BBN's HWIM (Hear What I Mean) systems (31). Neither Hearsay-II nor HWIM met the DARPA program's performance goal at its conclusion in 1976. However, the approach proposed by Hearsay II of using parallel asynchronous processes that simulate component knowledge sources in a speech system was a pioneering concept. The Hearsay II system extended sound identity analysis given the detection of lower-level information or evidence, which was provided to a global "blackboard" where knowledge from parallel sources was integrated to produce the next level of hypotheses. BBN's HWIM system, on the other hand, was known for its interesting ideas – which included a lexical decoding network

incorporating sophisticated phonological rules (aimed at phoneme recognition accuracy) – its handling of segmentation ambiguity by a lattice of alternative hypotheses, and the concept of word verification at the parametric level.

Another milestone of the 1970s was the beginning of a longstanding, highly successful group effort in large vocabulary speech recognition at IBM, in which researchers studied three distinct tasks over a period of almost two decades. Namely, these tasks included: the New Raleigh language (32) for simple database queries; the laser patent text language (33), for transcribing laser patents; and the office correspondent tasks called Tangora (34), for the dictation of simple memos. Finally, at AT&T Bell Labs, researchers began a series of experiments aimed at making speech recognition systems that were truly speaker independent (35). To achieve this goal, a wide range of sophisticated clustering algorithms were used to determine the number of distinct patterns required to represent all variations of different words across a wide user population. This research has been refined over a decade so that the techniques for creating speaker independent patterns are now well understood and widely used.

Research in the field of speech recognition in the 1980s was characterized by a shift in technology from the template-based approach to the statistical modeling method, most notably the HMM approach (36). The approach of HMM was well known and understood in only a select few laboratories, including: IBM, the Institute for Defense Analysis (IDA), and Dragon Systems, but it became more widely used in the mid-1980s. Today, most practical speech recognition systems are based on the statistical framework developed in the 1980s, and their results, with significant additional improvements, were achieved in the 1990s.

Another innovative technology that came into existence in the late 1980s was the method of applying a neural network to the problem of speech recognition (37). This approach was first introduced in the 1950s, but did not prove useful initially because of a number of practical problems (38).

The 1980s was a decade in which major emphasis was placed by the DARPA community on the development of a large vocabulary and a continuous speech recognition system. A significant research program was sponsored, which aimed at accomplishing high recognition accuracy for a 1,011-word database. Major research contributions resulted from efforts at CMU (also known as the SPHINX System) (39), which successfully integrated the statistical method of HMM with the network search strength of the earlier Harpy system.

In the 1990s, a number of innovations took place in the field of pattern recognition. The problem of pattern recognition, which traditionally followed the framework of Bayes and required the estimation of distributions for data, was transformed into an optimization problem involving minimization of the empirical recognition error (40). This fundamental paradigmatic change was caused by the recognition of the fact that the distribution functions for the speech signal could not be accurately chosen or defined, and that Bayes' decision theory becomes inapplicable under these circumstances. Fundamentally, the objective of a recognizer design should be to achieve the least recognition error, rather than to provide the best fit of a distribution function to the given (i.e., known) data set, as advocated by the Bayes criterion. The concept of minimum classification or empirical error subsequently

spawned a number of techniques, among which discriminative training and kernel-based methods such as support vector machines (SVMs) have become popular subjects of study (41).

During the 1990s, a key issue in the design and implementation of a speech recognition system was how to appropriately select the speech material used to train the recognition algorithm (42). A number of human language technology projects funded by DARPA in the 1980s and 1990s further enhanced progress in this regard, as shown by many papers published in the proceedings of the DARPA Speech and Natural Language/Human Language Workshop. These papers describe the development of accomplishments for speech recognition that were conducted in the 1990s (42) at Fujitsu Laboratories Limited.

In the 1990s, great progress was made in the development of software tools that enabled many individual research programs all over the world. As systems became more sophisticated (many large vocabulary systems involved tens of thousands of phone unit models and millions of parameters), a well-structured baseline software system was indispensable for further research and development, allowing for the incorporation of new concepts and algorithms. The system that was made available by the Cambridge University team (led by Steve Young), called the Hidden Markov Model Tool Kit (HTK) (43), was (and remains today) one of the most widely adopted software tools for research into ASR.

In the year 2004, Variational Bayesian (VB) estimation and clustering techniques were developed (44). The VB approach was based on a succeeding distribution of parameters. In 2005, Richardi (45) developed a technique to solve the problem of adaptive learning in ASR, and also proposed an active learning algorithm for ASR. In the same year, some improvements to the performance of large vocabulary continuous speech recognition systems were developed (46).

Furui (47) investigated a speech recognition technique that can adapt to speech variation using a large number of models, trained based on the clustering technique. In 2000, a 5-year national project "Spontaneous Speech: Corpus and Processing Technology" (48) was conducted in Japan. The collected corpus, "Corpus of Spontaneous Japanese" (CSJ) consisting of approximately 7 million words and corresponding to 700 hours of speech, was built, and various new techniques were investigated. These new techniques included flexible acoustic modeling, pronunciation modeling, sentence boundary detection, acoustic as well as language model adaptation, and automatic speech summarization.

To further increase the robustness of speech recognition systems, utterance verification and confidence measures are being intensively investigated (49). In order to have intelligent interactions in dialog applications, it is important to attach a number to each recognized event that indicates how confidently the ASR system can accept the recognized events.

In 2007, De Wachter (50) attempted to overcome problems of time dependency in speech recognition by using the straight-forward template matching method.

Sloin et al. (51) presented a discriminative training algorithm that uses SVM to improve the classification of discrete and continuous output probability with HMMs. The algorithm presented in their paper uses a set of maximum likelihood trained

HMMs as a baseline system, and an SVM training scheme to rescore the results of the baseline HMMs. Cui et al. (52) proposed techniques for automatically recognizing phonemes by using HMMs. The input of features into HMMs are extracted directly from a single phoneme rather than from a string of phonemes forming a word. Feature extraction techniques are also compared to their performance in phoneme-based recognition systems. They additionally describe a pattern recognition approach developed for continuous speech recognition.

2.2.3. Current and promising speech recognition technologies

Most current speech recognition systems use HMMs to deal with the temporal variability of speech, and Gaussian mixture models (GMMs) to determine how well the state of each HMM fits a frame or a short window of coefficient frames that represent the acoustic input. An alternative way to evaluate the fit is to use a feed-forward neural network that takes several frames of coefficients as input and produces posterior probabilities over HMM states as output. Deep neural networks (DNNs) that have many hidden layers and are trained using new methods have been shown to outperform GMMs on a variety of speech recognition benchmarks, sometimes by a large margin. New machine learning algorithms can lead to significant advances in ASR (53).

When neural nets were first used, they were trained discriminatively. Only recently have researchers shown that significant gains can be achieved by adding an initial stage of generative pre-training that completely ignores the ultimate goal of the system. This pre-training is much more helpful in deep neural nets than in shallow ones, especially when limited amounts of labeled training data are available. It reduces over fitting, and it also reduces the time required for discriminative fine-tuning with back propagation, which was one of the main impediments to using DNNs when neural networks were first used in place of GMMs in the 1990s. The successes achieved using pre-training led to a resurgence of interest in DNNs for acoustic modeling. Retrospectively, it is now clear that most of this gain comes from using DNNs to exploit information in neighboring frames and from modeling tied context-dependent states. Pre-training is helpful in reducing over-fitting, and it does reduce the time taken for fine-tuning, but similar reductions in training time can be achieved with less effort by careful choice of the scales of the initial random weights in each layer (53).

A DNN is a feed-forward, artificial neural network (ANN) that has more than one layer of hidden units between its inputs and its outputs. DNNs with many hidden layers are hard to optimize. DNNs with many hidden layers and many units per layer are very flexible models with a very large number of parameters. This makes them capable of modeling extremely complex and highly nonlinear relationships between inputs and outputs. This ability is important for high-quality acoustic modeling, but it also allows them to model spurious regularities that are an accidental property of the particular examples in the training set, which can lead to severe over-fitting. Weight penalties or early stopping can reduce over-fitting, but only by removing much of the modeling power (53).

ANNs trained by back-propagating error derivatives have the potential to much better learn models of data that lie on or near a non-linear manifold. In fact, two decades ago, researchers achieved some success using ANNs with a single layer of nonlinear hidden units to predict HMM states from windows of acoustic coefficients. At that time, however, neither the hardware nor the learning algorithms were adequate for training neural networks with many hidden layers on large amounts of data, and the performance benefits of using neural networks with a single hidden layer were not sufficiently large to seriously challenge GMMs. As a result, the main practical contribution of neural networks at that time was to provide extra features in tandem or bottleneck systems (53).

Over the last few years, advances in both machine learning algorithms and computer hardware have led to more efficient methods for training DNNs that contain many layers of non-linear hidden units and a very large output layer. The large output layer is required to accommodate the large number of HMM states that arise when each phone is modeled by a number of different "triphone" HMMs that take into account the phones on either side. Even when many of the states of these triphone HMMs are tied together, there can still be thousands of tied states. Using new learning methods, several different research groups have shown that DNNs can outperform GMMs at acoustic modeling for speech recognition on a variety of data sets including large data sets with large vocabularies (53).

There is a two-stage training procedure that is used for fitting DNNs. In the first stage, layers of feature detectors are initialized, one layer at a time, by fitting a stack of generative models, each of which has one layer of latent variables. These generative models are trained without using any information about the HMM states that the acoustic model will need to discriminate. In the second stage, each generative model in the stack is used to initialize one layer of hidden units in a DNN, and the whole network is then discriminatively fine-tuned to predict the target HMM states. These targets are obtained by using a baseline GMM-HMM system to produce a forced alignment.

While both DNNs and GMMs are nonlinear models, the nature of their nonlinearity is very different. A DNN has no problem modeling multiple simultaneous events within one frame or window because it can use different subsets of its hidden units to model different events. By contrast, a GMM assumes that each data point is generated by a single component of the mixture, so it has no efficient way of modeling multiple simultaneous events. DNNs are also good at exploiting multiple frames of input coefficients, whereas GMMs that use diagonal covariance matrices benefit much less from multiple frames because they require decorrelated inputs. Finally, DNNs are learned using stochastic gradient descent, while GMMs are learned using the EM algorithm or its extensions, which makes GMM learning much easier to parallelize on a cluster machine. Currently, the biggest disadvantage of DNNs compared with GMMs is that it is much harder to make good use of large cluster machines to train them on extensive data sets. This is offset by the fact that DNNs make more efficient use of data and so do not require as much data to achieve the same performance, but better ways of parallelizing the fine-tuning of DNNs is still a major issue (53).

Speech analytics is the process of analyzing recorded calls to gather customer information to improve communication and future interaction. The Large Vocabulary Conversational Speech Recognition (LVCSR) research was performed using the Switchboard (54) and CallHome (55) conversational telephone speech corpora (56).

Switchboard is a collection of approximately 2,400 two-sided telephone conversations between 543 speakers (302 males and 241 females) from all areas of the United States.

CallHome American English Speech was developed by the Linguistic Data Consortium (LDC), and consists of 120 unscripted 30-minute telephone conversations between native English speakers.

First, a HMM-GMM system was built using the Kaldi open source toolkit (57). The baseline recognizer had 8,986 sub-phone states and 200K Gaussians trained using maximum likelihood. The input features were speaker-adapted MFCCs. Additionally, there was a HMM-DNN system built by training a DNN acoustic model using maximum likelihood on the alignments produced by our HMM-GMM system (58).

The Kaldi toolkit provides several example pipelines for different corpora (57). The capabilities of these pipelines include linear discriminant analysis and maximum likelihood linear transform (LDA+MLLT), speaker adaptive training (SAT), maximum likelihood linear regression (MLLR), feature-space MLLR (fMLLR), and maximum mutual information (MMI, fMMI). GMMs and subspace GMM are also supported. Further, the training of DNN on top of GMMs involves layer-wise pre-training based on Restricted Boltzmann Machines, per-frame cross-entropy training, and sequence-discriminative training using lattice framework and optimizing the State Minimum Bayes Risk criterion (59). The training is of high computational expense, implementation and pipelines are optimized for parallel computing, and the training of DNNs supports the usage of GPUs to significantly speed up processing.

Comparing to Kaldi, HTK is the more difficult toolkit. Setting up the system requires the development of the training pipeline, which is time consuming and error-prone. The development of techniques, especially those beyond the tutorials provided, requires much more knowledge and effort than setting up the Kaldi system. Training techniques such as adaptation and discriminative training are possible, but the development of the toolchain is nearly impossible without expert knowledge. Compared to the other recognizers, the outstanding performance of Kaldi can be seen as a revolution in open-source speech recognition technology (60).

In recent years, some companies have presented very promising results of LVCSR systems. An 8.0% word-error rate (WER) on the Switchboard part has been presented by Saon et al. (61), and the main factors which influence the error rate are indicated in their paper. The performance of the individual networks as well as their score fusion combination is shown in Table 2.1 on the Hub5'00 test set (SWBD and CH parts).

Table 2.1. Comparison of WERs for CE and ST of CNN, DNN, RNN and various score fusions on Hub5'00

Model	WER SWBD		WER CH	
	CE	ST	CE	ST
CNN	12.6	10.4	18.4	17.9
DNN	11.7	10.3	18.5	17.0
RNN	11.5	9.9	17.7	16.3
DNN+CNN	11.3	9.6	17.4	16.3
RNN+CNN	11.2	9.4	17.0	16.1
DNN+RNN+CNN	11.1	9.4	17.1	15.9

Three types of models that differ in functionality and input features were used:

- regular DNNs with five hidden sigmoid layers;
- convolutional neural networks with two convolutional layers;
- partially unfolded recurrent neural networks, where the first hidden layer is recurrent and is followed by four non-recurrent layers.

Two experimental scenarios were considered: the first where cross-entropy (CE) training is used; and the second where 20–30 iterations of hessian-free sequence discriminative training (ST) were additionally applied (61). After the improved joint training of recurrent and convolutional nets, the WER was reduced to 9.3% (WER SWBD) and 15.6% (WER CH). The language modeling improvements enabled the achievement of the abovementioned 8.0% WER on SWBD and 14.1% WER with the CH speech corpus.

The following year, a collection of acoustic and language modeling techniques was presented that lowered the WER of the English conversational telephone LVCSR system to a record 6.6% on the Switchboard subset (62).

Microsoft's conversational speech recognition system is described by Xiong et al. (63), where recent developments in neural-network-based acoustic and language modeling were combined to advance the state of the art on the Switchboard recognition task. The combined system had an error rate of 6.2%. The main feature of this system was an ensemble of two fundamental acoustic model architectures – convolutional neural nets (CNNs) and long-short-term-memory nets (LSTMs), with multiple variants of each (63).

In 2017, the IBM research group dropped the WER from 6.6 (62) to 5.5 on the SWBD set (64). On the acoustic side, a score fusion of three models were used: one LSTM with multiple feature inputs, a second LSTM trained with speaker-adversarial multitask learning, and a third residual net (ResNet) with 25 convolutional layers and time-dilated convolutions. The training set of acoustic models was increased, and consisted of 262 hours of SWBD, 1,698 hours from the Fisher data collection, and 15 hours of CH audio (64).

The Microsoft 2016 conversational speech recognition system (63) was also updated in 2017: the resulting system reached a 5.1% WER on the 2,000 SWBD evaluation set (65). The full 2,000-hour corpus was used for the training of all neural networks. The acoustic model was enhanced by adding a CNN-BLSTM system (bidirectional LSTM).

The DNN method gives us quite precise results for the recognition of continuous speech with a large vocabulary, despite the fact that it requires high data resources which are complicated for low-resource languages. This idea underlines the need to search for an alternative method to DNN, which might be a combination of some recognition systems and technologies.

Artificially intelligent (AI) voice assistants are the new battleground between the big US tech companies, and while Google is no stranger – with voice search and Google Now having been available on Android smartphones for years – it was beaten into US and then UK households by Amazon and its Echo speaker (66). Amazon's Echo voice-controlled smart speaker was one of the first devices to use Amazon's voice assistant – a rival to Apple's Siri, Google's Assistant, and Microsoft's Cortana – which allows you to control music playback and more by simply speaking to it.

Google unveiled a number of new AI-driven products including Google Home (67), a voice-activated product that allows users to manage appliances and entertainment systems with voice commands, and which draws on the speech recognition technology used in its recently announced Google Assistant.

Cortana, the Microsoft phone assistant now built into Windows 10 (68), composes messages, performs searches, and sets calendar events by way of voice commands. It has been measured above 90% accuracy – quite an improvement considering Windows 95 had an error rate of close to 100%.

China's largest search engine, Baidu, has collected thousands of hours of voice-based data in Mandarin, which was fed into its latest speech recognition engine Deep Speech 2 (69). The system independently learned how to translate some Mandarin to English (and vice versa) entirely on its own using deep learning algorithms. In addition, the system is capable of "hybrid speech," something that many Mandarin speakers use when they combine English and Mandarin. Because the system is entirely data-driven, it actually learns to perform hybrid transcription on its own. This is a feature that could allow Baidu's system to transition well when applied across languages.

There has been a growing trend towards developing end-to-end systems which attempt to learn the separate components of ASR jointly as a single system over the last several years. This is valuable since it simplifies both the training and deployment processes. The two main approaches for this are the Connectionist Temporal Classification (CTC) and the attention-based sequence to sequence (seq2seq) models. CTC is a function that allows an RNN to be trained for sequence transcription tasks without requiring any prior alignment between the input and target sequences. Unlike CTC-based models, attention-based models do not have conditional-independence assumptions, and can learn all of the components of a speech recognizer – including the pronunciation, acoustics, and language model – directly. The performance of both models on the Hub5'00 benchmark is presented by Battenberg et al. (70). Without using a language model, attention models outperformed CTC models trained on the same corpus, but it was found that CTC models were significantly more stable, easier to train and ensured better recognition results if the language model was used.

2.2.4. Works on the recognition of the Lithuanian language

P. Kemėšis and L. Telksnys initiated investigations of language signals in Lithuania in the mid-1970s. Gradually, several groups formed to carry out scientific research in the field of language technology at: Kaunas University of Technology (KTU), Vilnius University (VU), Vytautas Magnus University (VDU), and the Mathematics and Informatics Institute (MII). This section briefly presents the activity of these institutions in the field of speech signal recognition along with their more significant results.

MII scientists realized a recognition system for separately pronounced Lithuanian numeral names (71). Using LP coefficients and DTW methods in the recognition experiments, a 1.9% word recognition error was achieved independent of the announcer, and 0.8% word recognition error for announcer-dependent recognition. Later, based on analytical expressions of the DTW method, the "Identification" recognition system was created for separate words (72). This was a program designed to monitor the recognition of isolated words. The system used the original methods for the identification of word boundaries and teaching, which allowed for increased recognition accuracy. Recognition studies of words pronounced separately continued, using HMM (73) and ANNs (74). In 2005, a modeled hybrid of ANN/HMM, based on a recognition system of separately pronounced words in the Lithuanian language (75), was presented. In recent years, MII presented the algorithm of speech signal segmentation to quasi-phonemes (76). Experimental studies have shown that the limits of quasi-phonemes marked by the system differed from those manually marked by approximately 23 ms.

Extensive research in the field of Lithuanian speech recognition has been performed by VDU, with much attention paid to HMM (77, 78). G. Raškinis examined the impact of various parameters in the HMM speech recognition system on solving the task of recognizing isolated words pronounced in the Lithuanian language of average volume, independent of the speaker. D. Šilingas examined sets of acoustic models and their properties in a recognition system for coherent Lithuanian speech. Evaluating the efficiency of Lithuanian phoneme sets, during which sets of graphemes and seven phoneme sets were compared, it was found that it is appropriate to include diphthongs and accent marks, but not to include information about the softness of consonants or to split mixed dialect diphthongs and affricates.

VDU scientists pay much attention to issues of language segmentation, which are based on logical teaching methods and are more present in certain works (79), as well as examining opportunities for the discrimination of voiceless explosive consonants according to the trajectories of the explosion and the following voice formants (80).

The KTU laboratory for speech signal study has lately, together with scientists from VU, paid a great deal of attention to speech synthesis issues, but previously many studies for speech recognition were also conducted.

Recently, the projection algorithm has been modified in a manner such that the standards have been depicted only using their phonetic transcriptions, and experiments with recognition of one announcer's voice command have been carried out (81, 82). Two sets of phonetic units for transcriptions were examined. In the first

case, 23 phonetic units (a slightly lower number than in the normal text, because affricates were identified with fricative sounds) of which 16 were assessed, where all the explosive consonants were marked with one symbol. Then, seven options were examined for phonetic training which differed by the overlapping of the phonetic contexts. Voice commands were selected completely at random. In the best case, 0.9% word recognition error was achieved.

An investigation into the recognition of Lithuanian voice commands based on multiple transcriptions was carried out in 2009(83).

The KTU laboratory for speech signal study paid the most attention to the discrimination of phonemes. Since 1985, classifiers (Euclid, Mahalanobis measures, and the dichotomous classifier were introduced, together with the optimization of feature space) have been compared, and the importance of phonemic discrimination in a particular context has been observed (84). Regularized discriminant analysis has also been used, in the hope of improving the automatic classification of phonemes. The dichotomous classifier, the Fisher classifier, and different features (autoregressive cepstral analysis, MFCC, recursive filters) have been used for comparisons. For tests, consonants *m, n* prior to three vowels, *a, u,* and *i* were selected, and were dictated by 20 speakers 10 times in the context of each vowel. At best, a 5.1% classification error of *m* and *n* phonemes was achieved (85).

Lately, the Šiauliai University has been involved in the research of Lithuanian speech recognition. Daunys (86) describes the features that allow the classification of phonemes based on the place of articulation. The possibility of creating methods for Lithuanian language segmentation and phoneme recognition is discussed. In another work involving the same author (87), the possibility of using visual information for speech recognition was examined in terms of creating decision trees and using them together with the differential features of phonemes.

In 2013, the INFOBALSAS project took place (88). High recognition accuracy was achieved for voice commands in a medical information system. The system was able to recognize the names of the most commonly encountered diseases, pharmaceuticals, and complaints in the medical practice by using an annotated speech corpus.

The Lithuanian Speech Managed Services Project (LIEPA) ended in 2015, and aimed at developing tools which would open the possibilities of working and communicating with computers and smart gadgets using Lithuanian speech, which presented numerous specific problems. The services developed are conducive to promoting students to use speech technologies, benefit or assist adults in various roles by enabling them to talk to computers in Lithuanian, help the disabled, and provide people with advice on how to correctly pronounce words in standard Lithuanian.

In 2015, Google announced a speech recognizer for the Lithuanian language. Google uses speech recognition in almost all of its products: Android OS, the Chrome browser, its search engine, and so forth. Because of this, languages such as English, Japanese, Russian, German, and others are recognized very well, as these languages have huge market potential and Google directs a lot of resources towards making their recognition better. For under-resourced languages, creating a good speech recognizer has limited market potential, and so creating a recognizer from scratch is not in

Google's interest. Perhaps a better way is to adapt acoustic models that have already been trained. As predicted, Google has recorded some training data and retrained one foreign language recognizer.

More Lithuanian speech recognition results and research results on the connection of recognizers are presented in Table 2.2.

Table 2.2. Speech recognition research results in Lithuania

Author	**Method**	**Speech corpus**	**Results**
Maskeliūnas (83)	Adapted language recognizer (English)	10 digit names (10 speakers, 20 utterances each digit)	RA 92.5%
Rasymas, Rudžionis (89)	Lithuanian Google recognizer	10 digit names (1,790 voice recordings)	RA 82.6%
Sipavičius, Maskeliūnas (90)	Lithuanian Google recognizer	Speech corpus of 42,515 voice records (238,885 phrases)	WER 40.74%
Laurinciukaite (158)	Word based HMM with fixed HMM states and number of Gaussian mixtures.	50 commands (31 speakers, 20 utterances each) phonetically annotated speech corpus.	RA 97.77%
	Phoneme based HMM	50 commands (31 speakers, 20 utterances each) phonetically annotated speech corpus	RA 93.91%
	Contextual phoneme based HMM	LRNO ~10 hours of Lithuanian radio broadcasts	RA 76%
Alumäe, Tilk (94)	TimeDelay Deep Neural Network	About 90 hours of Lithuanian tv speech recordings	WER 14.7%
Salimbajevas, Kapočiūtė-Dzikienė (95)	TimeDelay Deep Neural Network	6 hours of utterances from Seimas sessions	WER 21.3%
Gales et al. (93)	Triplone based HMM	10 hours of conversational Telephone Speech	WER 48.3%
Greibus et al. (96)	Triphone based HMM	LIEPA (46.56 hours of speech by 348 speakers)	CER 36.76%
Lileikytė et al. (97)	Triphone based HMM	Telephone speech, 40 hours	WER 42.4%
Raškinis et al. (98)	Recurrent neural network (RNN), BLSTM	50 hours of read speech, 50 speakers	PER 12.62
Pipiras et al. (99)	RNN, encoder–decoder-type models	Part of speech corpus LIEPA Isolated commands	Accuracy 0.993
	RNN, encoder–decoder-type models	Part of speech corpus LIEPA Long phases	Accuracy 0.992
Connection of several recognizers			
Rasymas, Rudžionis (131)	Connection of five recognizers: Lithuanian, Russian, English, and two German, with statistical classification methods	50 commands (drug names and names of diseases), 12 speakers, 20 pronouncements each (6,000 voice recordings)	RA 98.16%,

During the evaluation of Google's Lithuanian recognizer, a speech corpus containing 10 digit names was used. This corpus was gathered by recording the speech of random individuals, and every digit name was pronounced 1,790 times. Of these, 1,000 recordings were used for training, and 790 recordings were used for testing. The recognition accuracy of the Google Lithuanian recognizer was determined to be 82.6% (89).

Another experiment concerning the evaluation of Google's Lithuanian recognizer was conducted by Sipavičius and Maskeliūnas (90). This study involved 42,515 voice recordings (238,885 phrases), and the WER for all speech records that were processed by the Google speech recognizer was 40.74%, with a standard deviation at 37.70%.

Lithuanian is one of the development languages within the IARPA BABEL research program, and is therefore a test language in many papers that have studied low-resource training methods for speech recognition (91, 92, 93). For example, Gales, Knill, and Ragni (93) employed a simple approach for building graphemic systems for any language written in Unicode. The attributes for graphemes were automatically derived using features from the Unicode character descriptions. These attributes were then used in the construction of decision trees. This approach was examined with the IARPA Babel Option Period 2 languages, including the Lithuanian language. For each language, approximately 10 hours of Conversation Telephone Speech was distributed. A WER of just 48.3% was received for the Lithuanian language speech corpus using the HTK package.

Alumäe and Tilk (94) described the development of an automatic broadcast data transcription system for the Lithuanian language. The system performed fully automatic transcription of broadcast media recordings, including speech/non-speech detection, speaker diarization, speech-to-text conversion, and automatic punctuation restoration. The system was developed in collaboration with the Baltic Media Monitoring Group (BMMG). The Lithuanian large-vocabulary ASR system, developed by Tilde (95), is similar to this system and outperforms it.

The influence of the phoneme-set on the accuracy of Lithuanian speech command recognition was investigated by Griebus et al. (96). Four phoneme sets were discussed, and the LIEPA speech corpus was used for the training of an Acoustic Model. The phonetic representation of corpus transcriptions was generated by grapheme-to-phoneme transformation rules. Rule-based transformations for the Lithuanian language were proposed, and a recognition engine with the CMU Pocketsphinx decoder was used. Investigations using a 46-hour training speech corpus showed that a set of 36 phonemes showed the bests results – a command error rate (CER) of 3.76% using 6.78-hour speech corpus for testing.

Lileikyte et al. (97) presented a conversational telephone speech recognition system for the low-resourced Lithuanian language, developed in the context of the IARPA-Babel program. Phoneme-based systems and grapheme-based systems were compared to establish whether or not it is necessary to use a phonemic lexicon. Experimental results were reported for two conditions: Full Language Pack (FLP) and Very Limited Language Pack (VLLP), for which 40 and 3 hours of transcribed training data were available, respectively. Grapheme-based systems were shown to

give comparable results to phoneme-based ones. Including Web texts improved the performance of both the FLP and VLLP system – the best VLLP results, with a WER of 42.4%, were achieved using both Web texts and semi-supervised training.

The Lithuanian large-vocabulary ASR system, developed by Tilde, which is based on the open-source Kaldi toolkit (57), was presented by Salimbajevs and Kapociute-Dzikiene (95). The phoneme repository consisted of 29 grapheme-based phonemes, 1 unified filler-silence model, and 1 model for fragmented and out-of-vocabulary words. The ASR system used the Kaldi recipe for sequence discriminative training of TimeDelay Deep Neural Network (TDNN) acoustic models and iVectors for speaker adaptation. For the training of acoustic models, three speech corpora were used: a ∼100-hour Lithuanian speech corpus (52 hours of pure speech, 11,000 word forms, 61,000 utterances, 360 speakers), a ∼192-hour Seimas corpus (308,000 utterances), and a ∼20-hour dictated speech corpus (21,000 utterances). The developed ASR system for Lithuanian was evaluated using the standard WER metric on the following manually annotated test corpora: test_general – a 1-hour "general domain" set of audio segments from various radio and TV shows; test_seimas – a 6-hour set of randomly selected utterances from Seimas sessions; and test_lt_radio – a 2-hour set of audio segments from Lithuanian radio. A comparison with the Google Cloud Speech ASR and Alumäe&Tilk ASR (94) was then performed. The best results – 21.3% – were achieved on test_seimas, and the results of the same test for the other ASRs were worse: 28.4% for Alumäe&Tilk ASR, and 41% for Google ASR.

A survey of the research undertaken during the last 15 years to find an optimum mapping for Lithuanian ASR systems was performed by Raškinis et al. (98). This study also compared various phoneme- and grapheme-based mappings across a broad range of acoustic modeling techniques, including monophone- and triphone-based GMMs, speaker adaptively trained GMMs, subspace GMMs, feed-forward time delay neural networks (TDNN), and a state-of-the-art "low frame rate bidirectional long short-term memory" (LFR BLSTM) recurrent DNN. Experiments were based on a 50-hour speech corpus consisting of 50 speakers (25 males and 25 females) each reading book excerpts for approximately 1 hour. Full leave-one-out (or 50-fold) cross-validation was costly in terms of computational time, so an approximation was used. An open-source Kaldi ASR toolkit (57) was used for training and evaluating all ASR systems. Phone Error Rate (PER) criterion was used to compare the performances of different ASR setups. The results of speech corpus recognition varied, from 35.31% for monophone ASR to 12.62% for LFR BLSTM ASR. The best results were achieved using the phone 3-gram language model.

An ASR system for the Lithuanian language, which is based on deep learning methods and can identify spoken words purely from their phoneme sequences, was described by Pipiras et al. (99). Two encoder–decoder models were used to solve the ASR task: a traditional encoder–decoder model and a model with an attention mechanism. The performance of these models was evaluated in an isolated speech recognition task (with an accuracy of 0.993%) and a long phrase recognition task (with an accuracy of 0.992%), using part of the LIEPA speech corpus. Accuracy was calculated by adding the count of true positives and the count of true negatives, and

then dividing by the sum of the count of true positives, the count of true negatives, the count of false positives, and the count of false negatives.

2.3. Speech analysis methods and their characteristics

This section will discuss the methods for processing speech signals. The primary processing of speech signals consists of three stages: the initial filtration, the division of signal into frames, and the application of the window function. These three stages are used in almost all systems involving the recognition of speech and speaker, and many others besides.

Before performing the separation of speech features and their calculation, preparatory actions for speech signal processing are carried out: discretization of the signal, initial filtration, and windowing.

When performing the discretization of the speech signal, which aims to reduce the amount of data necessary for displaying the speech signal, the amplitude of the audio signal is measured and recorded many times per second. According to pre-defined goals, the maximum allowable values of the amplitude are determined. Depending on the number of values used for recording, the highest possible integer is assigned to the maximum value of the amplitude.

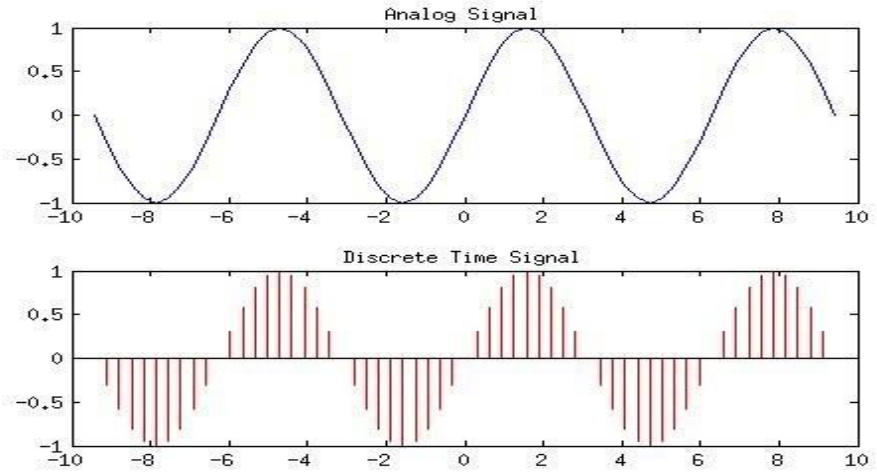


Figure 2.2. Analog and discrete signal

The frequency which is used to conduct measurements of the speech signal's amplitude is called the discretization frequency. Obviously, the higher the discretization frequency, the more the digital record of the speech signal corresponds to the analog (Figure 2.2). According to the Nyquist theorem, the discretization frequency must be at least twice as high as the maximum frequency of the signal recorded, in order to not lose important information in the signal. We have the discrete speech signal S, where the number of its discrete values is equal to N.

$$S=s(1), s(2),...,s(n),...s(N). \quad (1)$$

Most of the energy of the speech signal is concentrated within the area of low frequencies. As all frequencies are treated equally in the spectral analysis, this results in higher finite accuracy errors in further signal processing and, in addition, great variance in the estimates of some features. In order to avoid these shortcomings, so-called initial filtering (preemphasis) is carried out, which aims to remove the nonlinear frequentative distortions made during discretization. The aim of the initial filtration is to raise components of the higher frequency spectrum in order to increase their influence and to improve the quality of attributes used (100). In this way, components of the lower frequency spectrum are suppressed, and thus the spectrum is "leveled."

In terms of time, the initial filtration is carried out using a low order digital Finite impulse response (FIR) filter. The most commonly used first order FIR filter is defined as:

$$\tilde{s}(n) = s(n) - \alpha s(n-1), \tag{2}$$

here, $\tilde{s}(n)$ represents the signal filtered, *s(n)* the primary values of the discrete speech signal, and α the coefficient which determines the degree of leveling for the speech signal spectrum, selected from the range $0.9 \leq \alpha \leq 1.0$.

The filtered signal is broken into the sequence of *K* overlapping frames (windows). The windowing of the signal is defined by two parameters: the length of the window and the push, or step, of the window. The choice of length and step of the window depends on the methods used in the recognition system, but usually ranges (in length) from 10 to 30 ms, and the step of the window (overlapping) from 5 to 15 ms. This is done because it is assumed that within such a short interval of time the parameters of the human vocal tract fail to change (101), i.e., in such a short interval of time the human vocal tract can be described using permanent parameters. The overlap of the windows is used in order to more effectively use the information received from the two adjacent windows.

Each received window of the signal – in order to evaluate the continuity of the signal and the distortions made with the division – is multiplied by a certain window function, *v*:

$$\bar{s}(n) = s(n) \cdot v(n), 0 \leq n \leq N-1, \tag{3}$$

where *N* represents the window size in discrete values.

There are many potential functions of the window – such as rectangular, Hening, etc. – but for the most part the Hamming window (102) is used:

$$v(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right), 0 \leq n \leq N-1 \\ 0, other\ case \end{cases}. \tag{4}$$

The windows of discrete speech signal obtained are then used for the calculation of features, describing the analyzed units of the speech signal (words, syllables,

phonemes, diphones, triphones, etc.). These units are relatively independent of the individual properties of the announcer or the environment, and also independent of the report content (103). Algorithms are then used for the separation of the features – most often standard signal processing technologies such as digital filters, linear prediction, and spectral and cepstral analysis. These methods are reliable, and have been used for quite some time as they model the speech signal by combining it with the human auditory perception system (104). In addition, in the process of recognition, additional knowledge about the properties of the human vocal tract and the acoustic system are also used, allowing us to increase the accuracy of the speech recognition (103). It should be emphasized that there is no universal set of attributes yet discovered that would uniquely identify the fragment of the speech signal analyzed. All features distinguished have advantages and disadvantages.

The following subsections of this section analyze the methods used for the calculation of the features of the speech signal.

2.3.1. Linear prediction

One of the first methods to be used for digital analysis of the speech signals is linear prediction (105), during which the features calculated – coefficients of the linear prediction – can be used for analysis of the speech signals in speech recognition systems. The main idea of the coding method in linear prediction is that speech signal *Y* at moment in time *i* can be approximated with *p* speech signal values in the linear combination:

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \cdots + a_p s(n-p), \tag{5}$$

where the coefficients a_1, a_2, ... a_p in the analyzed window of the speech signal are considered stable. By adding the excitation member *Gu (i)* to the dependency 5 we get:

$$s(n) = \sum_{j=1}^{p} a_j s(n-j) + Gu(n) \tag{6}$$

where *u* represents the normalized excitation signal, and *G* the excitation coefficient.

The coefficients a_j represent linear prediction coefficients, for the calculation of which the Levinson–Durbin algorithm may be used (106). The TP method accurately models echoing speech signals (102). This is especially clear in quasi-stationary speech signal fragments, in which TP performs the accurate approximation of the coating in the signal spectrum generated by the vocal tract. In obtuse language fragments, the TP method is not so effective. Linear prediction requires fewer calculation resources than some other well-known methods (for example, the bank model of the digital filters).

2.3.2. Spectral analysis

The features of the speech signal obtained using spectral analysis are widely used. One of the main reasons for the wide prevalence of this method is that by performing spectral analysis we can attain important acoustic characteristics of the speech signal in various frequency bands.

As is already known, the speech signal is not a stationary thing, and the spectral analysis of speech signals is based on the assumption that the speech signal can be separated into short intervals where the signal becomes stationary or quasi-stationary (107).

For the spectral analysis of short intervals, two methods are used: the method of the filter banks, and the Fourier transformation algorithm.

Using the method of the filter banks, the speech signal *x (n)* is passed through the bank of filters made of *H* band filters (Fig. 2.3), which overlaps the range of the signal frequencies under study. In this way, the *M-1*-th band filter – the central frequency of which in the exit is the speech signal *x(n)* energy– and the energies of all the H filters approximate the short-term signal spectrum. The most commonly used filter banks are Mel or Bark scale, about which others have written extensively (108).

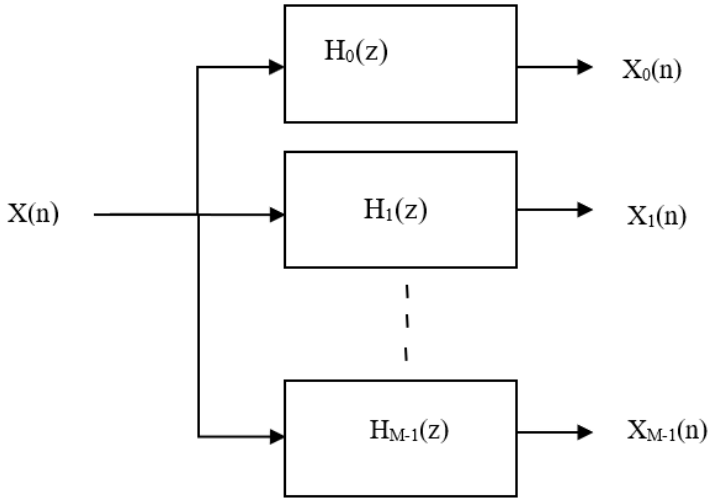


Figure 2.3. Multidimensional analysis filter banks

Another method of spectral analysis is the so-called fast Fourier transformation (FFT) (109). Its popularity was determined by the fact that the applications of the Fourier algorithm are much more convenient with the use of a computer, and it is much more easily realized than filter banks. The calculation of coefficients in the FFT is based on the discrete Fourier transformation equation:

$$S(l) = \frac{1}{N}\sum_{i=0}^{N-1} s(n)e^{-j\frac{2\pi}{N}ln}, where\ l = 0,1 \dots, N-1 \quad (7)$$

and the inverse discrete Fourier transformation:

$$(n) = \frac{1}{N}\sum_{l=0}^{N-1} S_n(l) e^{j\frac{2\pi}{N}ln}, where\ n = 0,1 \dots, N-1 \tag{8}$$

Formulas 7 and 8 can be used for the expression of the signal *s(n)* Fourier transformation in a short time interval by using a window function *v(n):*

$$S(l) = \frac{1}{N}\sum_{i=0}^{N-1} s(n)v(n)e^{-j\frac{2\pi}{N}ln}, where\ l = 0,1 \dots, N-1 \tag{9}$$

The resolution of the spectrum is inversely proportional to the length of the window *N*. Fourier transformation enables the conversion of the signal from the time scale into the frequency scale, and vice versa by using the inverse Fourier transformation (112).

2.3.3. Cepstral analysis

Lately, MFC (Mel-frequency cepstrum) features have become among the most widely used for speech recognition (110). Cepstral coefficients are used in speech recognition for a number of reasons. Firstly, from a theoretical point of view, a speech signal can be modeled as a convolution of several sources with different pulse characteristics (111). These sources of a signal include the vocal cords, mouth, throat, nasal cavity, lips, etc. Cepstral analysis allows us to distinguish these signal sources to perform deconvolution. It is assumed that the parameters of the signal sources distinguished must accumulate characteristics specific to separate sounds and individual speakers.

The second advantage is that the cepstral coefficients are less correlated with each other, and this greatly simplifies the process of further analysis.

Currently, the most common cepstral coefficients are calculated using the Fourier and discrete cosine transformation. Using this method, so-called Mel-frequency cepstral coefficients (MFCC) are calculated. The algorithm for MFCC calculation is presented in Figure 2.4, which consists of a number of stages. Firstly, the FFT is used to convert each frame of *N* samples from the time domain into the frequency domain. The scale of frequency is then converted from the linear to Mel scale. Then, the logarithm is taken from the results. In the final step, the log Mel spectrum is converted back to the time domain, resulting in the MFCC.

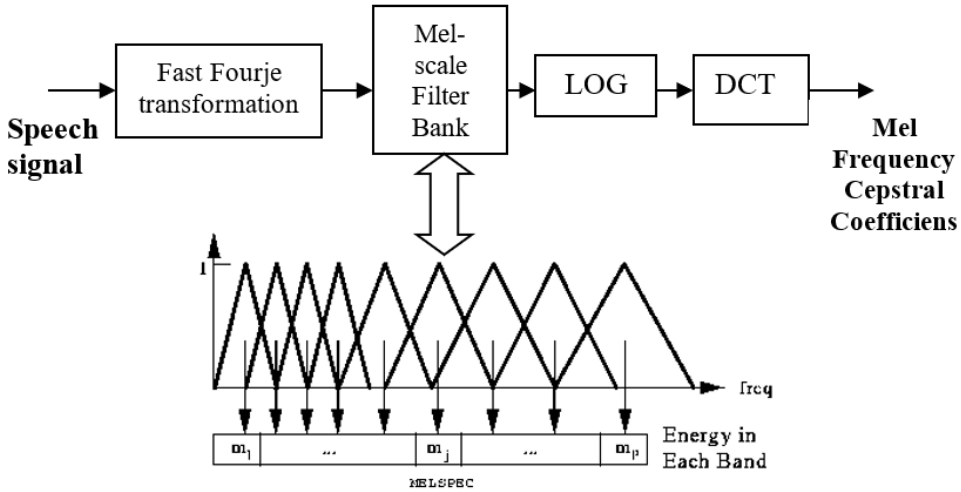


Figure 2.4. MFCC extraction and Mel-scale filter bank

First, using formula 9, the Fourier transformation coefficients of the speech signal are calculated, and the energy spectrum of the signal is obtained. The spectrum obtained is filtered using the Mel-frequency filter bank.

Other characteristics are also used in forming the vector of the features along with the MFCC or other coefficients described. Signal energy is commonly used for segmentation of the speech signals (112), when it is necessary to discern the speech signal from the noise because the energy of the speech signal is greater than the energy of the noise. The energy of the fixed-length discrete time signal can be expressed as:

$$E = log \sum_{n-1}^{N} s_n^2$$

. (10)

Often, the vector of features is supplemented by the dynamic cepstral coefficients, or so-called delta coefficients, which describe the rate of change of the cepstral coefficients. These delta coefficients helped to achieve better results in many works on ASR. Delta coefficients are calculated by the formula:

$$\Delta_k(l) = c_k(l) - c_{k-1}(l), \quad (11)$$

where *k* represents the window serial number.

It is observed that, in some cases, the rate of change of the cepstrum, speed coefficients (delta-delta coefficients), or the so-called acceleration coefficients can be useful. Delta-delta coefficients are calculated similarly to the delta coefficients, but with the subtraction of members of the vector in two adjacent delta coefficient features (104).

$$\Delta\Delta_k(l) = \Delta_k(l) - \Delta_{k-1}(l). \quad (12)$$

After the analysis of each window and the calculation of the features of the speech signal, the vector of the features – which will be used as an entrance for the ASR system – can be created from it:

$$X_k = (c_k \mid E_k \mid \Delta_k \mid \Delta E_k \mid \Delta\Delta_k \mid \Delta\Delta E_k). \quad (13)$$

Thus, the vector of the features can be formed of:

- 12 coefficients or the other cepstral, spectral, or linear prediction coefficients (c_k);
- 12 delta coefficients (Δ_k);
- 1 speech signal energy (E_k);
- 1 energy delta coefficient (ΔE_k);
- 12 acceleration coefficients ($\Delta\Delta_k$);
- 1 energy acceleration coefficient ($\Delta\Delta E_k$).

The number of elements in the feature vector depends on the number of MFCC coefficients used. For example, if we use 12 MFCC, then the vector of the features consists of 39 elements.

2.4. Methods for language recognition

Many methods of language recognition have been created, but only a few of them have proved worthy of use in recognition systems. We would distinguish three groups of language recognition methods:

- acoustic-phonetic methods;
- pattern recognition methods;
- artificial intellect methods.

In acoustic-phonetic methods, it is assumed that the language signal is of a finite duration, and consists of acoustically different phonetic units which are characterized by distinctive, temporal, and frequentative properties. After measuring these properties (usually the resonant frequencies, energy and amplitude levels, and zero-crossing number) and after the application of elementary rules (knowledge in acoustic phonetics, threshold values, and decision trees), the language signal is segmented and marked (phonetic transcription is assigned) – in this way realizing language recognition via the direct decoding of the signal into the transcription. However, segmentation is limited in the sense that it does not take into account the co-articulation phenomenon or the variability of sounds across different versions of words. Additionally, their classifications are quite primitive, and therefore acoustic-phonetic methods are not fit for purpose and are almost unused in modern recognition systems.

In pattern recognition methods, the variability of a language signal is evaluated using statistical methods (24). It is assumed that the signal is a random process which can be modeled. For the creation of the standards, statistical models are used, and

estimates of their parameters are found in the set of data used for training. In the recognition stage, the distance between the language signal examined and the reference models is evaluated using a statistical classification – usually Bayes' rule. The most common representatives of the statistical methods group are HMMs.

In artificial intelligence methods, the attempt is to imitate human language recognition: additional language skills are incorporated into the recognition process, and recognition systems are provided with the ability to adapt and learn. In the early methods used in expert systems, along with knowledge of acoustics, lexis, syntax, semantics, and even pragmatics were incorporated into the identification process. For the introduction of knowledge, the "top-down," "down-top," and board methods were used. Another group of artificial intelligence methods – indeed the largest – are networks of neurons. Networks of neurons imitate the human ability to learn from the data obtained by adjusting their perception according to it.

In the following subsections, the most widespread language recognition methods will be analyzed, including: HMMs, DTW, and neural networks.

2.4.1. Hidden Markov Models

HMMs are the most popular and among the most effective techniques used for speech recognition purposes today. Since their introduction in the 1970s, HMMs have been applied to a wide set of speech recognition tasks. Their popularity is caused both by the existence of effective training algorithms and by the existence of well-developed software tools, allowing researchers to quickly adapt them to their own tasks and purposes.

The HMM approach provides an entire framework, which includes an automatic supervised training algorithm with mathematically proven convergence properties (the Baum–Welch algorithm) and an efficient decoding scheme for recognition tasks (the Viterbi search algorithm). These models have the ability to generalize from large amounts of data by making structural assumptions that are reasonable for human speech and adjusting model parameters so as to optimize a meaningful objective function. The HMM theory is well described in the literature (e.g., 112, 113).

2.4.1.1. Basics and definitions of HMMs

HMMs are a well-known and widely used statistical method of characterizing the spectral properties of the frames of a pattern. These models are also referred to as Markov sources or probabilistic functions of Markov chains in the communications literature. The underlying assumption of the HMM is that a speech signal can be well characterized as a parametric random process, and that the parameters of the stochastic process can be determined (estimated) in a precise, well-defined manner. The HMM method provides a natural and highly reliable way of recognizing speech for a wide range of applications (102,114).

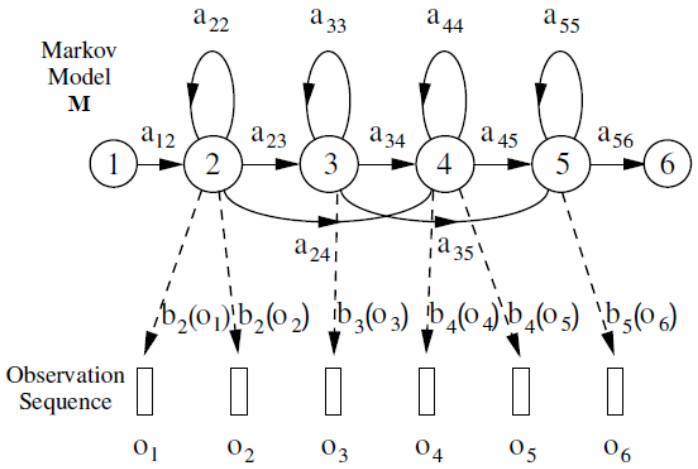


Figure 2.5. The Markov generation model

Figure 2.5 shows the typical structure of HMMs used in speech recognition. This model is called a left-to-right or a Bakis model, because the underlying state sequence associated with the model has a character such that as time increases the state index increases – that is, the system states proceed from left to right. Clearly, the left-to-right model exhibits a desirable property of being readily able to model speech with properties that change over time in a successive manner.

HMMs can be classified into discrete models or continuous models according to whether observable events assigned to each state (or transition) are discrete, such as code words after vector quantization, or continuous. Either way, the observation is probabilistic – that is, the model is a doubly embedded stochastic process with an underlying stochastic process that is not directly observable (it is hidden). Instead, it can be seen only through another set of stochastic processes that produce the sequence of observations.

An HMM for discrete symbol observations is characterized by the following:

$O = \{O_1, O_2, ..., O_T\}$ = observation sequence (input utterance) T length (duration) of observation sequence;

$Q = \{q_1, q_2, ..., q_N\}$ = (hidden) states in the model ;

N = number of states;

$V = \{v_1, v_2, ..., v_M\}$ = discrete set of possible symbol observations (VQ codebook);

M = number of observation symbols (VQ codebook size);

$A = \{a_{ij}\}$, a_{ij} = Prob(q_j at $t + 1$| q_i at t) = state transition probability distribution;

For the ergodic model, $a_{ij} > 0$ for all i, j. For the left-to-right model, $a_{ij} > 0$ for $i < j$;

$B = \{b_j(k)\}$, $b_j(k)$ = Prob(v_k at t| q_j at t) = observation symbol probability distribution in state j;

$\pi = \{\pi_i\}$, π_i = Prob(q_i at $t = 1$) = initial state distribution.

The compact notation $\lambda = (A, B, \pi)$ is used to represent an HMM. Specifying an HMM involves choosing the number of states, N, as well as the number of discrete symbols, M, and specifying the three probability densities of A, B, and 7r. This parameter set is calculated using the training data, and it defines a probability measure for $O = (O_1\ O_2\ \ldots\ O_T)$ – i.e., Prob $(0|\lambda)$, where each observation O_t is one of the symbols from V.

An observation sequence O is generated as follows:

Step 1: Set $t = 1$.

Step 2: Choose an initial state, i, according to the initial state distribution λ.

Step 3: Choose O_t according to b_i (k), the symbol probability distribution in state i.

Step 4: Choose j according to $\{ a_{ij}\}$ $(j = 1, 2, \ldots, N)$, the state transition probability distribution for state j.

Step 5: Set $t \leftarrow t+ 1$. Return to step 3 if $t < T$; otherwise terminate the procedure.

In the training phase, when 100 training utterances are used,

$$O^{(n)} = \left\{O_t^{(n)}\right\}_t^{T_n} \tag{14}$$

T_n = number of frames are obtained (n = 1, 2, … 100) λ*, which satisfies

$$\lambda^* = \text{argmax}_\lambda \prod_{n=1}^{100} Prob(O^{(n)}|\lambda) \tag{15}$$

is determined using the Baum–Welch algorithm (115). Here, Prob($O^{(n)}|\lambda$) indicates the conditional probability.

In the recognition phase for the unknown input, the probability that the observed sequence is generated from each HMM is computed, and the model with the highest accumulated probability is selected as the correct identification.

A pair, of model $m*$ and state sequence $q*$, (m*, $q*$), which satisfies

$$(m^*, q^*) = \text{argmax}_{(m,q)} Prob(O, q|\lambda_m), \tag{16}$$

is determined using the Viterbi algorithm, where λ_m is the mth model $\{m = 1, 2,..., M;$ M = vocabulary size), $O = O_1, O_2 \ldots O_T$ is input speech (T = number of frames), and q is a state sequence (22). Prob(O, q|λ_m) can be efficiently calculated using a forward-backward algorithm. These algorithms are precisely explained in the following subsections.

2.4.1.2. Three Basic Problems for HMMs

There are three key problems that must be solved when utilizing a HMM model.

Problem 1: Evaluation

Given the observation sequence $O = \{ O_1, O_2 \ldots O_T \}$ and the model $\lambda = (A, B, \pi)$, how can the observation sequence probability Prob(O|λ) be computed?

Problem 2: Uncovering Hidden State Sequence, given the observation sequence $O = \{ O_1, O_2 \ldots O_T \}$, how can a state sequence $I = \{ i_1, i_2 \ldots i_T \}$, which is optimal in some meaningful sense, be chosen?

Problem 3: Training

How can the model parameters $\lambda = (A, B, \pi)$ be adjusted to maximize Prob(O|λ)?

The principal structure of spoken word recognition systems based on HMMs is detailed in Figure 2.6. This structure requires the derivation of solutions to these three problems for particular use. The solution to Problem 1 is utilized to score each word model based on the given test observation sequence for recognizing an unknown word (116). The solution to Problem 2 is used to develop an understanding of the physical meaning of the model states. The solution to Problem 3 is employed to optimally obtain model parameters for each word model using training utterances.

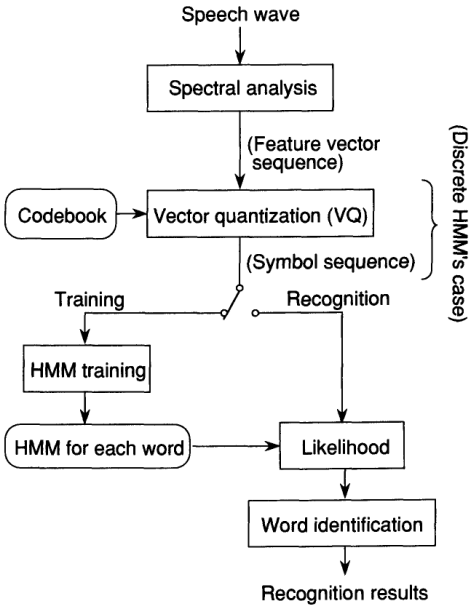


Figure 2.6. Principal structure of a word recognizer based on a HMM

Solution to Problem 1—Probability Evaluation

Prob(O|λ) can be represented as

$$P(O \mid \lambda) = \sum_{i_1, i_2, \ldots, i_T} \pi_{i_1} b_{i_1}(O_1) a_{i_1 i_2} \ldots a_{i_{T-1} i_T} b_{i_T}(O_T) . \qquad (17)$$

The summation in this equation is efficiently computed by the forward-backward procedure. Consider the forward variable $\alpha_t(i)$ as:

$$\alpha_t(i) = \text{Prob}(O_1, O_2 \ldots O_T, i_t = q_i|\lambda). \quad (18)$$

This indicates the probability of the partial observation sequence (until time *t)* and state q_i at time t, given model λ. We can solve for $\alpha_t(i)$ recursively as follows:

Step 1: $$\alpha_1(i) = \pi_i b_i(O_1) \qquad (1 \le i \le N) \quad (19)$$

Step 2*:* $$For\ T=1,2 \ldots T\text{-}1 \qquad (1 \le j \le N),$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}). \quad (20)$$

Step 3: $$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \quad (21)$$

This algorithm can be easily derived by transforming the HMM into a trellis or lattice diagram as shown in Figure. 2.7.

In a similar manner, a backward variable, $\beta_t(i)$, is defined as:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T | i_t = q_i, \lambda). \quad (22)$$

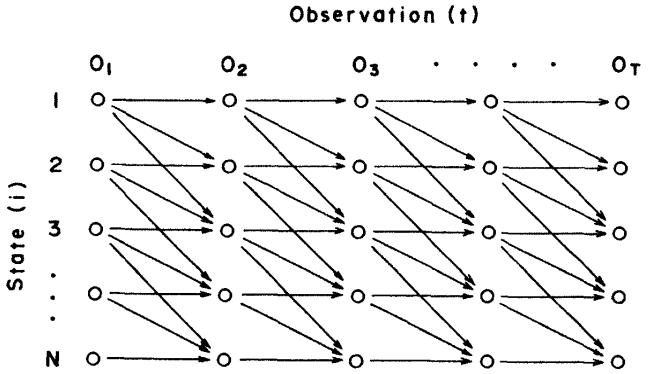


Figure 2.7. Trellis or lattice diagram representing an HMM

This demonstrates the probability of the partial observation sequence from $t+1$ to its conclusion, given state q_i at time t and model λ. Again, we can solve for $\beta_t(i)$ recursively as follows:

$$\beta_T(i) = 1 \qquad (1 \le i \le N), \quad (23)$$

for t = T-1, T-2 ... 1 $(1 \leq j \leq N)$,

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(O_{t+1}). \tag{24}$$

Then, $$P(O \mid \lambda) = \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_1(i). \tag{25}$$

Solution to Problem 2—Optimal State Sequence

Problem 2 can be solved using the Viterbi algorithm. This algorithm is similar to the forward-backward procedure, except that maximization over previous states is used in place of the summing procedure. The Viterbi algorithm is given as follows:

Step 1: Initialization

$$\delta_1(i) = \pi_i b_i(O_1) \qquad (1 \leq i \leq N) \tag{26}$$

$$\varphi_1(i) = 0 \tag{27}$$

Step 2: Recursion

For $2 < t < T$, $1 < j < N$,

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij} \right] b_j(O_t) \tag{28}$$

$$\varphi_t(j) = \arg\max_{1 \leq j \leq N} [\delta_{t-1}(i) a_{ij}] \tag{29}$$

Step 3: Termination

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \tag{30}$$

$$i_t^* = \arg\max_{1 \leq i \leq N} [\delta_T(i)] \tag{31}$$

Step 4: State sequence backtracking

For $t = T - 1, T - 2, \ldots 1$,

$$i_t^* = \varphi_{t+1}(i_{t+1}^*) \tag{32}$$

Here, P^* is the maximum likelihood, and φ indicates the maximum likelihood state sequence. If one only wishes to compute P^*, φ values need not be maintained. The Viterbi algorithm is a form of the well-known dynamic programming method.

In the Viterbi algorithm, the observation probability at each state is usually converted to a logarithmic value. Then, the accumulated probability can be quickly calculated by using the DP method with only maximum selection and summation calculations. That is, for $1 < t < T$, $1 < j < N$:

$$\delta_t = \begin{cases} \log \pi_i \\ \max\limits_{1<i\le N}[\delta_{t-1}(i) + \log a_{ij} b_j(O_t)] \end{cases} \qquad (2 \le t \le \mathrm{T}) \qquad (33)$$

is calculated, and finally the log-likelihood:

$$P^{*'} = \max_{1\le i\le N} \delta_T(i) \qquad (34)$$

is obtained. Since the logarithmic values are used, the dynamic range of the accumulated values becomes small, and therefore there is no need to be concerned about the underflow problem.

Along with the development of the HMM, the fundamental DP technique is now often called the Viterbi algorithm.

Solution to Problem 3—Parameter Estimation

An iterative procedure, such as the Baum-Welch method, or a gradient technique for optimization is used in solving this problem. With the Baum–Welch algorithm, $\xi_t(i,j)$ is first defined as:

$$\xi_t(i, j) = P(i_t = q_i, i_{t+1} = q_j \mid O, \lambda) . \qquad (35)$$

This denotes the probability of a path being in state q_i at time t and making a transition to state q_j at time $t + 1$, given observation sequence O and model λ. $\xi_t(i,j)$ can be written as:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)} . \qquad (36)$$

In the above equation, $\alpha_t(i)$ accounts for the first t observations, ending in state q_j at time t. The term $a_{ij}b_j(O_{t+1})$ accounts for the transition to state q_j, at time $t + 1$ with the occurrence of symbol O_{t+1}. The term $\beta_{t+1}(j)$ accounts for the remainder of the observation sequence. Prob(O|λ) is the normalization factor.

Next, $\gamma_t(i)$ is defined as:

$$\gamma_t(i) = P(i_t = q_i \mid O, \gamma) . \qquad (37)$$

This represents the probability of being in state q_i at time t, given observation sequence O and model λ. $\gamma_t(i)$ can be expressed as:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)}. \tag{38}$$

$\gamma_t(i)$ can be related to $\xi_t(i,j)$ by summing $\xi_t(i,j)$ over j, giving:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j). \tag{39}$$

If $\gamma_t(i)$ and $\xi_t(i,j)$ are each summed over the time index t (from $t = 1$ to $t = T-1$), quantities are obtained which can be interpreted as:

$\sum_{t=1}^{T-1} \gamma_t(i)$ = expected number of transitions made from q_i,

and:

$\sum_{t=1}^{T-1} \xi_t(i, j)$ = expected number of transitions from state q_i to state q_j.

Using these quantities, the HMM parameter values can be reestimated such that:

$$\tilde{\pi}_i = \gamma_1(i) \qquad (1 \le i \le \mathrm{N}), \tag{40}$$

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{41}$$

$$\tilde{b}_j(k) = \frac{\sum_{t=1, O_t=k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}. \tag{42}$$

The reestimation formula for π_i corresponds to the probability estimation of being in state q_i at $t = 1$. The reestimation formula for a_{ij} represents the ratio of the expected number of transitions from state q_i, to q_j divided by the expected number of transitions out of state q_i. Finally, the reestimation formula for $b_i(k)$ is equal to the ratio of the expected number of times of being in state j and observing symbol k, divided by the expected number of times of being in state j.

It can be verified that Prob(O|λ^*) ≥ Prob(O|λ) (λ^*= π^*, A*, B*). Therefore, if λ^* is iteratively used in place of λ and the above reestimation calculation is repeated, the probability of *O* being observed from the model can be improved until a limiting point is reached.

The above reestimation algorithm is generally called the EM algorithm, since it consists of the iterations of expected value calculation and likelihood maximization.

2.4.1.3. Continuous Observation Densities in HMMs

All of the discussion thus far have only considered when observations were characterized as discrete symbols chosen from a finite alphabet, and therefore a discrete probability density within each state of this model can be used. However, these observations are usually continuous signals or vectors, with possibly serious degradation associated with this discretization. Hence, it would be advantageous to be able to use HMMs with continuous observation densities to model continuous signal representation directly.

The most general representation of the model probability density function (pdf), for which a reestimation procedure has been formulated, is a finite mixture of the form:

$$b_j(O) = \sum_{k=1}^{M} c_{jk} \mathrm{N}(O, \mu_{jk}, U_{jk}), \qquad (1 \le \mathrm{j} \le \mathrm{N}) \quad (43)$$

where O is the observation vector being modeled, c_{jk} is the mixture coefficient for the *k*th mixture in state *j*, and N is any log-concave or elliptically symmetrical density (e.g., Gaussian). Usually, a Gaussian with mean vector μ_{jk} and covariance matrix U_{jk} for the Ath mixture component in state j is used as N. The mixture gains c_{jk} to satisfy the stochastic constraint:

$$\sum_{k=1}^{M} c_{jk} = 1, \qquad (1 \le \mathrm{j} \le \mathrm{N}) \quad (44)$$

$$c_{jk} \le 0, \ 1 \le j \le N, \ 1 \le k \le M, \quad (45)$$

so that the pdf is properly normalized, i.e.:

$$\int_{-\infty}^{\infty} b_j(O) d) = 1, \qquad (1 \le \mathrm{j} \le \mathrm{N}). \quad (46)$$

It can be shown that the reestimation formulas for the coefficients of the mixture density are of the form:

$$\tilde{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{k=1}^{M} \gamma_t(j,k)}, \tag{47}$$

$$\tilde{\mu}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) O_t}{\sum_{t=1}^{T} \gamma_t(j,k)}, \tag{48}$$

$$\tilde{U}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) * (O_t - \mu_{jk})(O_t - \mu_{jk})'}{\sum_{t=1}^{T} \gamma_t(j,k)}, \tag{49}$$

where prime denotes vector transpose, and where $\gamma_t(j,k)$ is the probability of being in state j at time t with the kth mixture component accounting for O_t, i.e.:

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jk} \mathrm{N}(O_t, \mu_{jk}, U_{jk})}{\sum_{m=1}^{M} c_{jm} \mathrm{N}(O_t, \mu_{jm}, U_{jm})} \right]. \tag{50}$$

2.4.2. The Dynamic Time Warping Method

The DTW classifier is assigned to the group of methods for comparison of samples, and uses dynamic programming – i.e., it compares samples, minimizing the distance between them (117).

Suppose you have a standard $A = \{a1, a2 \ldots a_R\}$ and the unknown sample $T = \{t1, t2 \ldots t_Z\}$, where a_i is the i-th vector of the standard features and t_j the j-th vector of the unknown sample. In terms of the geometric interpretation, the indexes of the standard and sample vectors are arranged in order to form a grid, the size of which is $R \times Z$. Each grid point (i, j) defines the distance $d(i, j)$ between the i-th standard vector and the j-th vector of the unknown sample (in our case, Euclidean distance is used). Using the grid formed, we can find the warping trajectory $W = \{w1, w2 \ldots w_K\}$, where $w_k = (i_k, j_k)$, and maximum $(R, Z) \le K \le R + Z - 1$. In order to find the optimal warping trajectory (the DTW distance between the samples is examined), the DTW algorithm solves the task of minimizing by determining the trajectory that minimizes the distance between the considered samples (97):

$$d_{DTW}(A,T) = min\frac{1}{K}\sum_{k=1}^{K} w_k. \quad (51)$$

By increasing the number of samples examined, the number of possible trajectories exponentially increases. Dynamic programming finds an optimal trajectory, summing up the distance between the samples examined and the minimum distance of the trajectory starting at the point (1,1) and ending at (*i, j*):

$$D(i,j) = d(i,j) + min\begin{cases} D(i-1,j-1) \\ D(i-1,j) \\ D(i,j-1 \end{cases}, \quad (52)$$

where *D(i, j)* represents the total distance calculated for the trajectory starting at point (1,1) and ending at *(i, j)*, and *d (i, j)* is the distance between the *i*-th and the *j*-th vectors of the samples examined. The number of possible trajectories is large, so a warping trajectory has some key limitations (98):

1. Monotony and consistency. Trajectory points must be monotonic, as transitions are possible only to the next points:

$$0 \le i_k - i_{k-1} \le 1, \;\; 0 \le j_k - j_{k-1} \le 1. \quad (53)$$

2. Restrictions of the ends. The beginning and the end of a warping trajectory are indicated. In the simplest case, the starting point is (1,1) and end (R, Z) – i.e., the trajectory begins at the first grid point and ends at the last.

3. Local restriction of the direction. The search number of warping trajectories can be limited in determining the number of possible movements in one direction. The work uses local restriction of the Itakura direction (Fig. 2.8), where *P* is the sequence of possible transitions described with coordinates $P \rightarrow (p_1,q_1)(p_2,q_2) \ldots (p_R,q_R)$.

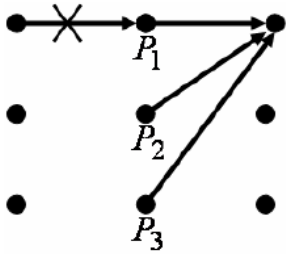


Figure 2.8. Local restriction of the Itakura direction, $P_1 \rightarrow (1,0)$, $P_2 \rightarrow (1,1)$, $P_3 \rightarrow (1,2)$. Consecutive transitions are not available

4. Global restriction of the direction. In addition, the search number of wiggle trajectories can be limited in defining the search area. Limitations of the left and right are expressed as (Fig. 2.9):

$$1 + (D(i) - 1) / Q_{\max} \le D(j) \le 1 + Q_{\max} (D(i) - 1), \quad (54)$$

$$Z + Q_{\max} (D(i) - R) \le D(j) \le Z + (D(i) - R) / Q_{\max}, \quad (55)$$

where Qmax is the maximum coefficient of deflection.

Often, the restriction of the Itakura parallelogram is used with $Q_{\max} = 2$, see Figure 2.9.

$$R = Q_{\max} (Z - 1) + 1, \quad (56)$$

$$Z = Q_{\max} (R - 1) + 1. \quad (57)$$

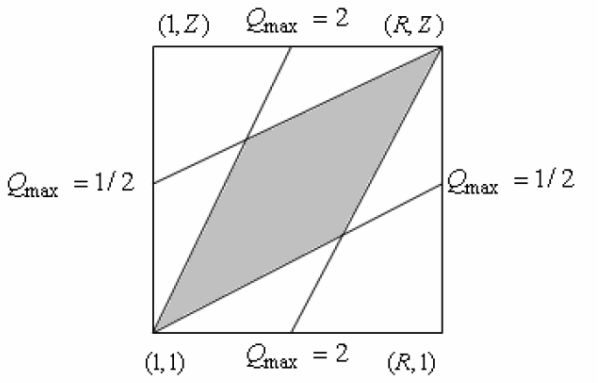


Figure 2.9. Global restriction of the direction of the Itakura parallelogram, Qmax = 2

The complexity of the algorithm in the DTW recognition system is $O(R^2)$, (N = 2R, where N is the number of all vectors). The error of the DTW classification can be found by dividing the number of samples which are tested and incorrectly classified, E, by the number of total test samples, Z:

$$DTW_{KL} = \frac{E}{Z}. \quad (58)$$

2.4.3. The method of artificial neural networks

The neuron model consists of a number of inputs that are summed by multiplying them by certain coefficients, called weights, and are directed to the activation function (118, 119). The neuron model is expressed by the following formula:

$$y = f(h) = \left(\sum_{d=1}^{D} w_d x_d + w_0 \right) \quad (59)$$

where y represents the neuron output value, x_d the elements of the input vector, w_d the elements of the weight vector, w_0 the neuron threshold, and $f(.)$ the activation function. A zero input of $x0$ was introduced which is usually constant – i.e., $x_0 = 1$ – and which uses such activation functions as a threshold, sigmoid, and hyperbolic tangent.

The simplest neural network is the perceptron, which is composed of a single layer of K neurons connected to the D input. Each perceptron output yk is determined by the input $x_1, x_2 \ldots x_D$ function, which is calculated using the following formula:

$$y_k = f(h_k) = f\left(\sum_{d=0}^{D} w_{kd} x_d\right), 1 \le k \le K. \quad (60)$$

In the perceptron learning process, weights are changed in a manner such that the network output vector would be as close as possible to the vector of desired values $t_1, t_2 \ldots t_K$. The error function is expressed by the formula:

$$EF = \frac{1}{2}\sum_{k=1}^{K} (t_k - y_k)^2 \quad (61)$$

The simplest method for perceptron teaching is the delta rule, which aims to minimize perceptron output error: another amendment is performed after each weight correction iteration, which is proportional to a derivative of the loss function according to all of the weight vector components (120):

$$w_{kd}(t+1) = w_{kd}(t) + \Delta w_{kd}(t), \quad (62)$$

$$\Delta w_{kd}(t) = \eta(t_k - y_k)x_d, \quad (63)$$

where η is the learning speed parameter, using which the perceptron learning speed is regulated. The classification error can be found dividing the number of vectors which are tested and incorrectly classified, E, by the number of vectors tested, Z:

$$TK_{KL} = \frac{E}{Z}. \quad (64)$$

2.5. Hybrid approach technologies

The term *hybrid approach* could be understood to mean the incorporation of several different recognition algorithms or methods. The basic assumption underlying the hybrid approach is that different recognition methods are able to extract and process different kinds of information present in the acoustic signal, and if they were used together this could lead to an overall increase in the accuracy and robustness of recognition. It should be noted that in many current state-of-the-art speech recognition systems, hybrid recognition principles are implemented one way or another. For example, some speech recognizers work using features of MFCC, while others work

in parallel using features of PLP, and several HMM-based recognizers are used with different training and most likely acoustic state search strategies implemented (121). In the case of Lithuanian voice command recognition, the hybrid approach is also important because it may potentially enable the use of foreign-language-trained speech recognition engines, adapted to recognize Lithuanian commands with the proprietary Lithuanian speech recognizer. Foreign language recognizers should allow for the exploitation of large amounts of acoustic data used to train these recognizers (for economic reasons, there is not and probably never will be such an amount of Lithuanian acoustic data as would be required to train speech recognizers, and as exists for such languages as English or Spanish). Earlier experiences with the adaptation of foreign language speech engines to recognize *modeling of Lithuanian* has shown that it is possible to achieve very high recognition accuracy for many Lithuanian voice commands using only the appropriate selection of their phonetic transcription (88). Such an approach enables us to make the development of limited vocabulary applications easier and more economically viable.

However, it has become clear that not all of the voice commands that are necessary for some successful voice-based services can be recognized equally well using an adapted recognition engine. Whilst a proprietary Lithuanian speech recognizer may potentially better deal with some acoustic situations that are not present in other languages, it is necessary to develop specific acoustical models. It is also necessary to use a proprietary recognizer to recognize problematic voice commands well enough. The need to combine the results provided by two different recognizers requires the implementation of the hybrid approach.

The problem of how to combine different recognizers still remains largely unsolved, and requires further research. Various methods have been proposed to combine recognition results obtained from different sources. The most popular method is the method called heteroscedastic discriminant analysis (122). However, before finding the most efficient ways to combine the hypotheses produced by various recognizers, a number of other questions should first be resolved. Among those problems, some issues remain, such as: the possibility of attaining complementary information from different speech recognizers; defining when and in which contexts foreign language recognizers could be used, and when it is necessary to use purely Lithuanian acoustic models; and finding the limits and possibilities of adapting foreign language speech engines to recognize Lithuanian voice commands (123).

2.5.1. The connection of recognition methods

In order to eliminate the weaknesses of ANNs and HMMs in solving problems of speech recognition, some scientists began to integrate them into continuous hybrid architecture. The main goal of hybrid systems was to use the positive features of HMMs and ANNs to increase the reliability and flexibility of speech recognition systems. Many different architectures and new training algorithms were proposed, (75, 124) and in this section we will briefly review the main trends in the application of these hybrid systems.

One example of the use of hybrid systems is the use of more efficient training algorithms with discrimination characteristics in evaluating the probabilities of HMM

states (125, 126). Using the vectors of acoustic observations, the neural network is trained so that its outputs perform the probability evaluation of nonparametric continuous HMM state transitions. The essence of this method is the idea that, instead of the standard Viterbi algorithm, the error spreading back algorithm is used, which is characterized by stronger discriminatory properties and, therefore, allows us to improve the accuracy of the recognition system.

Neural networks are used in discrete HMMs as the vector quantization algorithm. Instead of the standard clustering algorithms, self-organizing neural networks can use the much wider and more varied range of means, and thus carry out more efficient quantization of vectors. Examples of such hybrid systems can be found in a number of papers (127, 128).

Another area for the use of hybrid systems is the general optimization of the recognition system. Generally, both ANNs and HMMs are trained separately, although there are works (129) in which training is carried out in parallel for both technologies simultaneously. In these works, ANNs are used for transforming the vector of acoustic features to more efficient vectors of observations in order to optimize the parameters of the hybrid system models.

There have been attempts to use hybrid ANN/HMM systems in the recognition of the Lithuanian language. Filipovičius (75) analyzed the application of the hybrid speech recognition method in the recognition of separately pronounced Lithuanian language words, taking into account the acoustic properties of the Lithuanian language. The recognition system for separately pronounced Lithuanian language words was modeled and tested in his research, and a comparison of the efficiency of this system with the efficiency of a HMM system was carried out. Although the recognition accuracy of both systems was very similar (90.5% for HMM, 90.7% for hybrid), the hybrid system used significantly fewer parameters in order to achieve this result.

2.5.2. The connection of several recognizers

The method of combining multiple speech recognizers by using voting and language model information with ROVER seeks to reduce WERs for ASR by exploiting differences in the nature of the errors made by multiple speech recognizers (130). Rover proceeds in two stages: first, the outputs of several speech recognizers are aligned and a single word transcription network (WTN) is built. The second stage consists of selecting the best scoring word (with the highest number of votes) at each node. The decision can also incorporate word confidence scores if these are available for all systems.

An example of the connection of several recognizers is presented the work of Rasymas and Rudžionis (131). Attempts were made to adapt several foreign-language (English, Russian, and two German) speech recognizers for the recognition of a limited Lithuanian vocabulary, and to evaluate some (*k*-nearest neighbors, linear discriminant analysis, quadratic discriminant analysis, logistic regression, and maximum likelihood) methods used for the combination of different speech recognizers. One native (Lithuanian) recognizer was also used.

The method to combine recognizers was performed as follows:

- the voice command was passed to all speech recognizers in parallel;
- each recognizer then produced an output;
- the output of the recognizers formed the hypothesis, i.e., the score of how well the audio signal matched the acoustic model;
- this hypothesis score was then passed to a classification algorithm, which made the final decision (131).

Rasymas and Rudžionis used a speech corpus of 25 drug names and 25 disease names, gathered by recording the speech of 12 people (5 females and 7 males). Each of these speakers pronounced each command name 20 times at a sampling rate of 16 kHz in a single session. It should be noted that the corpus used in these experiments was part of the larger Lithuanian speech corpus of medical terms. The selection of this particular set of voice commands was based on the fact that 25 commands were those voice commands which resulted in the highest number of recognition errors using a proprietary Lithuanian speech recognizer, while the additional 25 commands were selected randomly.

The results of this study showed that the highest accuracy was obtained when the k-nearest neighbors method was used with 15 nearest neighbors. In this case, 98.16% accuracy was achieved (131).

In another paper by the same authors (132), the CART classifier was used for the combination of speech recognizers. Using this classifier, a 97.58% average recognition accuracy was obtained. Comparing the results of this experiment with those discussed earlier, (132) it is evident that using the CART classifier produced results that were 0.58% less accurate than those produced using the 15-nearest-neighbor classifier.

The same authors again conducted another similar experiment for creating a hybrid speech recognition system using recognizers from four foreign languages: recent Google recognizers in Russian, English, and two in German were used, alongside one for the Lithuanian language (133). A main speech corpus containing 10 names of digits was used. The corpus was gathered by recording the speech of random people, and every digit name was pronounced 1,790 times. In total, 1,000 recordings were used for training classifiers, and 790 recordings were used for testing. Some recordings were not recognized by any recognizer, in which case these recordings were omitted from further training and testing (133).

The highest result of 97.51% accuracy was acquired when all foreign language recognizers and the Naïve Bayes classifier was used. The result achieved by using all five recognizers alone was 96.69% (133).

2.6. Speech corpus

One of the most important elements in a voice recognition system is a well-crafted speech database, or speech corpus. In order to collect a proper speech corpus, a number of resources are required. The biggest problem in this regard is one of human resources – speakers. In order to have a speech corpus that is versatile, a large number of recordings is required from different speakers, and one that is composed of individuals of different sexes, with different dialects, from different regions, etc.

2.6.1. Corpus development trends

The first known speech corpus system – TI-DIGITS – was created in the United States in 1984, and involved isolated digit names and sequences collected by Texas Instruments (134).

Since that time, a lot of speech corpora have been created, and have come to be distinguished by their size, function, and the detail of their annotation. The needs of telephony have encouraged the development of speech corpora containing sequences of digits or commands. Along with creating mobile-user speech corpora in different environments – quiet work offices, standing, moving vehicle environments, etc. – specialized speech corpora have also been collected, where speakers follow texts from radio and television broadcasting.

Some speech corpora occupy a lot of storage space, especially those that are specialized for the study of certain peculiarities of speech: generic speech and dialects; men, women, children; free talk and reading; individual words, teams, and coherent text. A universal speech corpus is intended to reveal any general characteristics of speech. Speech corpora can be composed of annotated sentences, words, syllables, and sounds.

Some key trends in the development of speech corpora can be noted:

1) applied purpose speech corpora are steadily increasing in number, as their creation is financed by large telephone companies, the automobile industry, and the combined forces of science and business;

2) there is a growing number of national corpora, documenting language as a monument to a nation's social and cultural environment;

3) speech corpora are increasingly becoming more well-annotated, and designed for detailed scientific research work.

The robustness of a recognition system is heavily influenced by its ability to handle the presence of background noise. A first attempt to compare the performance of different algorithms was made using the Noisex-92 database (135). This consists of recordings from one male and one female speaker – from a vocabulary of English digits – that have been distorted by artificially adding background noise at different signal-to-noise ratios (SNRs) and in different noise conditions. A database suitable to obtain comparable recognition results for the speaker-independent recognition of connected words in the presence of additive background noise and for the combination of additive and convolutional distortion has been developed by Pearce and Hirsch (136). In their work, a selection of eight different real-world noises were added to speech over a range of signal to noise ratios. The noise signals were added to the clean TIDigits database (137) at SNRs of 20, 15, 10, 5, 0, and −5 dB. Noises were recorded in different places: a suburban train; a crowd of people; a car; an exhibition hall; a restaurant; a street; an airport; and in a train station. The average degradation of word accuracy was more than 25% when SNR was changed from 5 to 0 dB (136) during testing using an HTK-based recognizer.

2.6.2. The development of a speech corpus in Lithuania

For the decades that Lithuania has been involved in research on speech recognition, researchers have mostly used databases of audio recordings to solve small, specific tasks. Most language technology researchers in Lithuania carry out their works at Kaunas University of Technology, the Institute of Mathematics and Informatics of Vilnius University, and at the Vytautas Magnus University.

At the Kaunas University of Technology's Language Research Laboratory, ASR research has been carried out since 1980, and the laboratory has developed a command sequence and digital speech corpus. In creating Lithuanian computer dialogues, the University has accumulated and improved the Lithuanian spoken language corpus LTDIGITS (138) to such an extent that it is comparable to the USA TI-DIGITS corpus (139).

Vilnius University's Institute of Mathematics and Informatics has accumulated a Lithuanian news radio corpus – LRNO. The institute, along with partners, is creating a program of voice managed services (for example: a Lithuanian language neologisms pronouncer, browser, and controller, that allows online information search and the management of some computer tasks by voice) in accordance with the Lithuanian language information society program for 2009–2013. The institute is also improving the technologies and tools of the spoken language, including by developing a text reader, a command and phrase recognition engine, and a Lithuanian speech recognition engine.

Vytautas Magnus University has accumulated a universal spoken Lithuanian language speech corpus, and research is ongoing into spoken Lithuanian language autodivision. Automatic spoken transcription of the Lithuanian language is also being created, involving a smaller special corpus collected for language learning – for example, the SACODEYL young people spoken language corpus (140).

2.6.3. Annotation of speech corpora

The transcription of speech recordings at phone-level is a fundamental task in phonetics and speech technology research. The identification of phone segments in speech material is the starting point for many studies. Typically, this is done manually, but an accurate fully-manual approach may require as much as an 800-fold increase in real time – i.e., up to 13 hours for a one-minute recording (141). This processing time is a major drawback for manual labeling, especially when faced with a very large spontaneous speech corpus.

As was mentioned above, low-resource languages typically have a low presence on the internet, with limited textual resources in electronic form and little available knowledge regarding the language (142). The Lithuanian language sits among other low-resource languages because there is no annotated and transcribed acoustic training data for it: the collection, transcription, and annotation of speech data are typically expensive and time-consuming tasks (142). The key results of one paper which presents a review on ASR for under-resourced languages (143), show that some European languages are still considered under-resourced (for speech processing, the

following languages are mentioned: Croatian, Icelandic, Latvian, Lithuanian, Maltese, and Romanian).

2.7. The specific properties of Lithuanian phonetics

The main specific properties of phonetics are as follows: a rich set of phonemes, including diphthongs, dialect diphthongs, and affricates; the features of the phonemes, including the length of vowels, the softness of consonants, and assimilation; and a complex accentuation system. We will review each of these in detail below.

2.7.1. The issue of phonemic set

Typically, the basis of acoustic modeling units in speech recognition systems is a set of speech phonemes. Later, specific algorithms based on data can automatically create models of contextual phonemes from simple models of phonemes. The set of phonemes in the Lithuanian language is significantly larger than that of the English language, and even linguists themselves disagree on a defined set of phonemes. Some distinguish special phonemes – affricates – whilst others disagree, arguing that an affricate is only a combination of two other phonemes. Disputes also arise as to whether mixed dialect diphthongs should be modeled as separate phonemes. Therefore, research into recognition techniques for Lithuanian speech does not have an agreed upon standard of how to select an initial set of phonemes for acoustic modeling, and instead often uses different sets of phonemes. This raises a number of problems, namely: the failure to reuse trained acoustic models; and the difficulty of comparison between the results of experiments carried out independently because it is unclear to what extent each was influenced by the set of phonemes used, how many signs were selected, and how many detection methods were used.

Phonemes in Lithuanian language phonetics have the following specific characteristics:

1) specific phonemes are distinguished, i.e., diphthongs and mixed dialect diphthongs;
2) the vowel can be long or short;
3) each consonant may be hard or soft depending on the vowel following it;
4) consonants become similar (assimilate) in the junction with other consonants.

The Lithuanian language has many dialect diphthongs in which the two phonemes are closely related, and linguists offer to model this how they would a single phoneme. The two main classes of dialect diphthongs are: 1) diphthongs, which consist of two vowels – ai, au, ei, eu, ie, uo; 2) and mixed dialect diphthongs, which consist of combinations between the vowels a, e, i, and u, and the consonants called semivowels – 1, m, n, and r. Dialect diphthongs are pronounced as a single unit, and are quite different from other pairs of phonemes which do not form dialect diphthongs. Another reason why it is suggested to model dialect diphthongs as one phoneme is that dialect diphthongs can be stressed, and stress can be short, with a strong start, or with a strong end. In the latter case, the stress falls on the consonant, which is a part of the dialect diphthong, and individual consonants cannot be stressed.

Lithuanian language vowels can be either long or short. The longitude of Lithuanian vowels is defined by the following rules:

if the vowel is accented by right or curly stress, then it is long;
if the vowel is accented by left stress, then it is short;
if the vowel is unstressed, then it is short;
all long and nasal vowels – ą, ę, į, ų, y, ū – are long;
the vowel ė is always long.

Consonants in the Lithuanian language are hard or soft, depending on the context. The softness of consonants in the Lithuanian language can be described by the following rules:

- if, after the consonant, there is the vowel a, ė, or o, then the consonant is hard.
- if, after the consonant, there is the vowel e, i, u, or a sign of softness, then the consonant becomes softer;
- if, after the consonant, there is a soft consonant, the consonant become softer – softness is a transitive property, which is transmitted in reverse;
- the consonant j is always soft.

In the phonetics of the Lithuanian language, coarticulation effects are possible, the most common of which are:

- The homogenization (assimilation) of two consonants, for example, a**tb**ėgti, u**žs**iūti.
- The connection of two similar consonants at the junction into a single – for example, i**šš**okti, u**žs**akyti.
- The softness or hardness of a consonant is dependent the on vowels that follow it – for example, ka**ma**na, ka**im**enė.

2.7.2. The complex system of accentuation

The Lithuanian language has a specific accentuation system. An accented syllable can have an accent of rising frequency called a *circumflex*, or an accent of falling frequency called an *acute*. The accent of rising frequency can be either short or long. Therefore, in the Lithuanian language, three versions of accent are available overall. It should also be noted that the accent of changeable parts of speech can jump from one syllable to another and change the type of accent, depending on the changeable form. For the changeable parts of the language, some stress paradigms are distinguished that specify the rules on how the stress changes depending on the changeable form. The stress may be the only feature allowing us to distinguish two different words, for example, šáuk (šauti – infinitive) and šaũk (šaukti – imperative). Therefore, it is necessary to model the stress, and to include it in the sets of phonemes and dictionaries for the pronunciation of recognizable words. Since the stress can change the forms of words, a tool for automated stressing which could be integrated into the speech recognition system is necessary.

2.7.3. Design of Lithuanian SAMPA

SAMPA (Speech Assessment Methods Phonetic Alphabet) is a machine-readable phonetic alphabet (144). SAMPA essentially consists of a mapping of the

symbols of the International Phonetic Alphabet (IPA) onto ASCII codes in the range 33–127, the 7-bit printable ASCII characters. Associated with the coding (mapping) are guidelines for the transcription of the languages to which SAMPA has been applied.

Lithuanian SAMPA must meet the following requirements (145):

Phonetic resolution. Phonetic resolution refers to the ability of SAMPA to distinguish between and assign different codes to the allophones of the same phoneme. The greater the phonetic resolution of an alphabet, the greater the coverage of SAMPA's potential applicability.

Standard Lithuanian has a few phonological features that are responsible for the vast majority of the allophonic variations of Lithuanian phonemes:

- Lithuanian consonants may be palatalized or non-palatalized;
- Lithuanian vowels may be short or long;
- Lithuanian syllables may be stressed in several different ways.

Traditional phonetic transcription of Lithuanian uses three diacritic marks for modeling syllable accentuation: grave (ˋ), acute (ˊ), and circumflex (˜). Grave and acute diacritic marks are used for indicating a sharply falling accent. The grave mark is traditionally placed over a short-stressed vowel and over the first element of a semi diphthong if it represents one of the short vowels. A circumflex diacritic mark indicates a smoothly rising accent.

Readability. Readability refers to how naturally and easily SAMPA-based transcriptions can be read by humans. Readability is very important, as phonetic transcriptions of speech corpora are manually verified and corrected by humans during the iterative corpus validation stages. For best readability, SAMPA codes must be kept similar to the symbols used by the traditional Lithuanian spelling.

Traditional Lithuanian spelling is based on the set of 32 symbols that includes 9 diacritic symbols: a, ą, b, c, č d, e, ę, ė, f, g, h, i, į, k, l, m, n, o, p, r, s, š, t, u, ū, ų, v, z, and ž. Thus, SAMPA must define ASCII codes for substituting diacritic symbols. Secondly, Lithuanian orthography is essentially morphphonological – i.e., standardized spelling reflects essential phonological changes but tolerates phonological inaccuracies as well. There are some Lithuanian sounds represented by digraphs – i.e., uo, ch, dz, and dž.

2.8. Speech recognition over the telephone

Using telephony applications, a user can check their bank balance via telephone or receive an automated call from their doctor's office reminding them of their next appointment (146). Speech Server provides tools for developing applications that run over the telephone, or telephony applications. Speech Server applications can possess the following capabilities:

- Speech recognition allows users to respond to application prompts;
- Touch-tone capabilities, called dual-tone multi-frequency (DTMF), let users respond to application prompts via the telephone keypad;
- Text-to-speech (TTS) capabilities allow applications to read and speak written text to users;

- Speech servers, such as MSS or IBM WebSphere Voice Server, provide ASR and TTS resources which are the basis of the speech interface. A special program placed in the server runs the dialog between human and computer (147).

There is quite a large variety of voice telecommunication systems, but, as they are already well established, they can be broadly distributed into two groups:

- Interactive voice response (IVR);
- Spoken language interface (SLI).

The IVR system query is performed with the help of the input of dual-tone multi-frequency (DTMF) tones via keypad, while the response is presented by playing pre-recorded phrases or by synthesizer (148).

MSS is an IVR system, integrated with Visual Studio 2005. One of the features of MSS'2007 is VoIP support. VoIP essentially allows users to place and receive queries over the Internet. Speech servers can accept VoIP queries without any additional software or hardware. An effective dialogue is the key component to a successful interaction between a voice-only application and a user. A voice-only application interacts with the user entirely without visual cues. The dialogue flow must be intuitive and natural enough to simulate two humans conversing. It must also provide the user with enough context and supporting information to understand the next action step at any point in the application.

Speech servers integrate a whole set of computer interaction means: voice, computer, telephony, internet, and databases. It has been noted that, for example, MSS'2007 is the basis for thousands of different demo applications. These applications have very high efficiency (as the cost of service and transaction time is reduced in orders) (149).

The typical structure of voice dialogue implemented in MSS'2007 involves a main or initial prompt that is played on entry, plus a number of other supporting prompts that either restate the question or directive in a contextually appropriate way, or offer help as the user traverses the state (Fig. 2.10).

Successful recognitions proceed to the next state. A give-up or failure can either send the user back to some predetermined state in the system to try another approach, or the system may offer to connect the user to a live operator (149).

One more feature of MSS'2007 is how it allows *barge-ins*. This means that users can provide their input before the prompt has finished speaking. Experienced users will be able to get to the part of the application that they need more quickly, whereas new users will find it helpful to say the selection they want as they hear it. By allowing *barge-ins*, users will complete their calls more quickly, saving them time and saving the operator resources – as shorter call times mean fewer resources required (149).

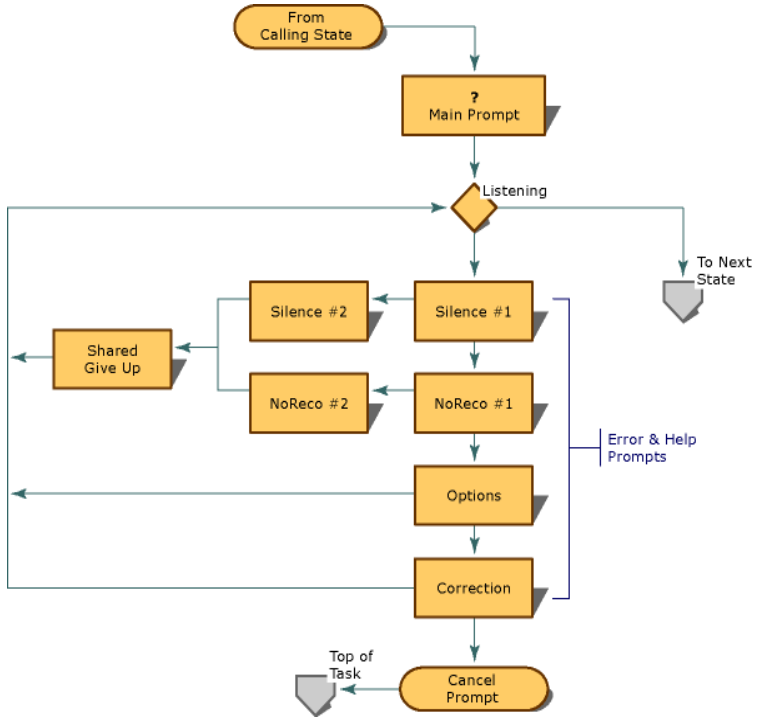


Figure 2.10. Typical structure of voice dialogue in MSS'2007

It is common in industries that have recently entered the telecommunications industry to refer to an automated attendant as an IVR. The terms, however, are distinct, and mean different things to traditional telecommunications professionals. Emerging telephony and VoIP professionals often use the term IVR as a catch-all to signify any kind of telephony menu, even a basic automated attendant; the term voice response unit (VRU) is sometimes used as well. MSS'2004 only supported the markup language of language applications (speech application language tags – SALT), whereas the new version supports three types of projects:

- SALT W3C standard language for Web and telephony;
- VoiceXML: VoiceXML and SALT describes the integration of internet, telephone, and language technology;
- Voice Response Workflow: unlike SALT and VoiceXML, this allows for the visualization of the progress of queries of applications.

It also includes a number of new tools:

- International Grammar Builder and Grammar Design Advisor: the Conversational Grammar Builder allows for the quick and easy creation of grammar in natural conversation. You can choose to build grammar in Grammar XML (GRXML) or the Visual Studio Editor Grammar. The Grammar Design Advisor also provides warnings about possible incorrect grammar;
- Lexicon Editor: Allows for adding or changing the pronunciation of words (over Conversational Grammar Builder);

- Pronunciation Editor: Allows for adding or changing the pronunciation of grammar (applies only to the Grammar Editor);
- Analysis and tuning;
- Business Intelligence Tools.

Speech server has two main components: Speech Engine Services, and ASP.NET. Speech Engine Services (SES) has two components: the Speech Recognizer engine, and the Speech Synthesis Engine (6).

The speech recognition engine's semantic markup language (SML) creates an XML document, and this document contains words or phrases which are recognized by the voice recognition engine. There is also a numerical value that indicates the degree of reliability of the user-pronounced word or phrase, as defined in the program's grammar.

MSS'2007 is an American (US) product, but it can be used to create applications for other languages. MSS'2007 fully supports the English (United Kingdom), English (United States), French (Canada), German (Germany), and Spanish (United States) languages. Unlike the 2004 version, MSS'2007 uses a UPS (Universal Phone Set) transcription system (150).

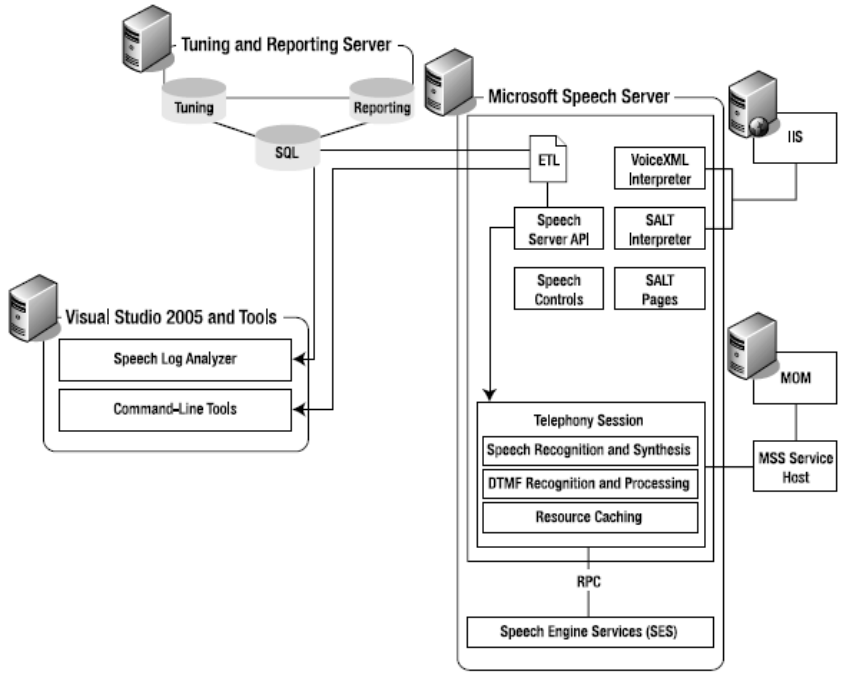


Figure 2.11. MSS's interaction with servers and Visual Studio.

2.9. Chapter summary

1. The analysis of the literature shows that the most widely applied method for the recognition of isolated word commands is the HMM. For the recognition of

isolated commands, HMMs that are based on words, phonemes, or contextual phonemes are applied. The energy of the signal and cepstral coefficients of the Mel scale and their first and second order derivatives are generally used as the recognition features. The length of the feature vector is 39. DNNs have an advantage over HMMs, but additional resources are required – financing, in particular – for their use in Lithuania.

2. The basic idea behind the hybrid approach is that different recognition methods are able to extract and process different kinds of information present in the acoustic signal, and their joint use could lead to an overall increase in recognition accuracy and robustness. Filipovičius (75) used the hybrid recognition system based on the combination of the ANN and HMM methods, but the accuracy in recognition of both systems was very similar (HMM – 90.5%, and hybrid – 90.7%). Rasymas and Rudžionis (131) obtained better results by connecting five parallel acting recognizers, (with a recognition accuracy of 98.16%), but, in this case, the structure of the recognition system became more complex.
3. One of the most important elements of a voice recognition system is a well-crafted speech database, or speech corpus. In order to collect a proper speech corpus, a number of resources are required. The biggest problem in this regard is one of human resources (speakers). In the absence of a speech corpus annotated at the phonemic level, a word-based HMM should be applied for the recognition of isolated commands, because the straightforward segmentation of a speech corpus at the word level is more suitable for HMMs. The set of phonemes in the Lithuanian language is significantly higher than in the English language. Therefore, research into recognition techniques for Lithuanian speech does not have a generally agreed-upon basis for how to select an initial set of phonemes for acoustic modeling, and instead uses different sets of phonemes.
4. Speech servers integrate a whole suite of means of computer interaction, including: voice, computer, telephony, internet, and databases. MSS'2007 is an IVR system, providing tools for developing applications that run over the telephone, or telephony applications.
5. For the recognition of codes, speech corpora should be collected that consist of Lithuanian digits names and Lithuanian names.
6. A very high digit-recognition accuracy is required to ensure the sufficient recognition accuracy of a sequence of digits. In 2015, Google announced a speech recognizer for the Lithuanian language, but two experiments with this recognizer showed that it does not provide a high recognition accuracy for Lithuanian digits names or words. Therefore, new methods of recognition for Lithuanian digits names should be created that allow for the attainment of a recognition accuracy for digits of above 99%.

3. RESEARCH TECHNIQUE AND INSTRUMENTS

3.1. International Classification of Diseases

The International Classification of Diseases (ICD) is the international standard diagnostic tool for health management, epidemiology, and clinical purposes (151). This system is designed as a suite of categories to permit the systematic recording, analysis, comparison, and interpretation of morbidity and mortality data collected internationally. The ICD is a major project that aims to statistically classify diagnoses of diseases, symptoms, complaints, and other health disorders from their medical names into an alphanumeric code which permits the retrieval, analysis, and easy storage of data.

The basic ICD is a single-coded list of three-character categories, each of which can be further divided into up to 10 four-character subcategories. In place of the purely numerical coding system of previous revisions, the Tenth Revision (ICD-10) uses an alphanumeric code with a letter in the first position and a number in the second, third, and fourth positions. The fourth character follows a decimal point. Possible code numbers therefore range from A00.0 to Z99.9 (151). As an example, A69.21 represents the code for Meningitis due to Lyme disease.

3.2. The use of an adapted language recognizer for Lithuanian voice commands

Adapted language (multilingual) voice recognition is based on a particular language (usually a more commonly used one) which has pre-developed models of acoustic phonetic units in the system, thus making it available to use for the recognition of another language (usually a less commonly used one) (152).

Using an adapted language model for Lithuanian language recognition can be divided into two tasks: first, establishing principles, such as transcribing target text and ensuring that it will be acceptable to the chosen adapted language recognition program; secondly, accumulating the resources (speech corpus) and software tools necessary in order to secure Lithuanian language recognition research and development (153).

Non-Lithuanian language recognizers (e.g., German, English, Spanish) should be compatible with phonetic symbols (e.g., UPS, IPA), by which the same word might be written in several different forms (transcription) that are interpreted equally, semantically.

Transcription can be defined as language elements (sounds, phonemes) or objective phonetic notes (recording) “specially” written, using artificial tools or others. Alphabets take letters and diacritical marks, such as the Lithuanian word recording of another language recognizer’s “understandable” SAPI symbols. Multiple transcription occurs when the same voice command recognition system uses several word (or phrase) transcriptions that have the same semantic meaning.

So far, the only voice server application for the Lithuanian language is achieved by using foreign language transcriptions for Lithuanian words. A very good recognition accuracy for Lithuanian digit names was attained by using the Microsoft English (U.S.) v6.1 recognizer (99.8% for a female speaker) (147). This led to the

conclusion that using speech server for the recognition of Lithuanian digit names would achieve high recognition accuracy results. Unfortunately, the results of recognition experiments have shown that IPA transcriptions are not suitable for speech server (147). Therefore, UPS type transcriptions should be used for MSS'2007 speech server (150).

There are examples of a module that was created for one language being applied to another (152). For this purpose, linguistic or acoustic experience is used (153). Accurate research into English recognizer applications being used for the recognition of Lithuanian last names and Lithuanian digit names are presented in multiple sources (83, 154). The use of other language recognition tools for the Lithuanian language is based on transcribing Lithuanian words or phrases into another language, for example, English. By using IPA transcriptions, the Lithuanian word "nulis" could be automatically transcribed into English as "n uh l ih s." Then, these symbols could be recognized by English recognizer.

MSS'2007 was chosen for preparing telephony services, as it performs speech recognition, speech synthesis, and telephony control operations. For creating new programs, Microsoft Visual Studio 2005 was used. Voice output can be performed from the synthesized text, from processed audio files, or may be derived from the synthesized files and pre-prepared mixture audio files. MSS'2007 has four language recognizers to choose from: German (Microsoft Speech Recognizer 9.0 for MSS (German-Germany)), English (Microsoft Speech Recognizer 9.0 for MSS (English-US)), French (Microsoft Speech Recognizer 9.0 for MSS (French-Canada)), and Spanish (Microsoft Speech Recognizer 9.0 for MSS (Spanish-US)).

The test for measuring the accuracy of the recognition of voice commands was prepared in MSS'2007 (Fig. 3.1): the speech dialog component "answerCallActivity1" answers an incoming call; "questionAnswerActivity1" asks the question and receives the user's answer; "gotoActivity1" jumps to another component; and "disconnectCallActivity1" disconnects an existing call. Such a framework is suitable for testing the recognition of Lithuanian voice commands by a selected speech recognizer. The prompt, grammar, and target properties of the questionAnswerActivity1 and gotoActivity1 speech dialog components should be defined before the testing procedure begins.

In debugging mode, the testing program presents the recognized word transcription along with the confidence measure: the word is considered as recognized if the confidence measure is above 0.2.

3.3. Speech corpora used in the studies

In order to investigate recognition accuracy in relation to voice commands, it is first necessary to have a properly prepared speech corpus containing Lithuanian number and name voice commands from many speakers. Voice commands were dictated and recorded using the "inp_sr16.exe" program in the MS DOS operating system environment.

The speech corpus for Lithuanian digit names was named SKAIC30. This speech corpus was formed of utterances from 30 different speakers – 23 females (F) and 7 males (M). Each announcer dictated the Lithuanian digit names from zero to

nine 20 times at a sampling rate of 16 kHz, 16-bit, saved in WAV format. Dictation was performed in a non-isolated room, and no additional signal processing was performed. No artificial noise was added to the signal because the main goal of the investigation was to assess the hybrid technology.

The features of the speech corpus used for the recognition of letters are presented in Table 3.1. The LETTERS and NATO speech corpora were used to verify the possibilities of letter recognition. The LETTERS speech corpus consists of Lithuanian letters pronunciations – for example, the letter "m" is pronounced as "em," and so on. The NATO alphabet is the most widely used spelling alphabet (155). The final choice of code words for the letters of the NATO alphabet, and for the digits, was made after hundreds of thousands of comprehension tests involving 31 nationalities. The qualifying feature was the likelihood of a code word being understood in the context of others (155).

Table 3.1. Features of speech corpora used for letter recognition

Speech corpus	Number of words	Number of speakers	Number of utterances
LETTERS	26	2 (1M,1F)	50
NATO	26	2 (1M,1F)	50
NAMES1	250	2 (1M,1F)	20
NAMES2	70	10 (5M,5F)	20
NAMES3	26	21 (9M,12F)	20

The NAMES1 speech corpus consisted of the utterances of approximately 10 Lithuanian names for each letter, and was used during the first step of the Lithuanian name selection procedure which is described in section 3.4. When the best-recognized Lithuanian names were determined, the second speech corpus – NAMES2 – was prepared, consisting of 2 to 3 Lithuanian names for each letter. The NAMES3 speech corpus consisted of utterances of 22 Lithuanian names and 4 words (kju, Wašington, iksas, ygrekas), which represent 26 letters used in disease codes. The NAMES3 speech corpus consists of voice recordings of 21 speakers – 12 females and 9 males.

The voice commands of both SKAIC30 and NAMES3 were dictated 20 times (utterances). The dictations were performed in a quiet environment, but not in professional sound recording studio, and digital recording equipment and a professional microphone were used. When dictating, it was very important to highlight the beginning and the end of the utterance.

In order to ensure the quality of sound material recorded, after the dictation of all commands repeated checks were carried out to search for records that were mistakenly dictated. Any errors were then corrected: noises were eliminated, and vaguely pronounced commands were newly dictated.

The SKAIC30 speech corpus of Lithuanian digit names consists of 6,000 different voice records, and the NAMES3 speech corpus of names consists of 10,920 voice records.

The INFOBALSAS project ended in 2013. The main goal of the project was to develop hybrid voice command recognition technology and implement it in the first

practical informative service using the recognition of Lithuanian voice commands. The informative service was oriented towards the workplace of the physician/pharmacist, and sought to support and hasten the search for information in a pharmaceutical data base. In total, 731 voice commands from a medical speech corpus (complaints and the names of drugs and diseases) were used in the construction of a hybrid recognizer. Each voice command was pronounced by 12 different speakers 20 times, and there were 175,440 commands overall. Of the complaints – e.g., "pilvo skausmas," "regėjimo sutirikimai," "žemas kraujo spaudimas" – 81.73% were phrases. Disease names were 72.35% phrases (e.g., "gerklės skausmas," "padažnėjęs širdies plakimas"). The names of drugs – e.g., Betalok ZOK, TerraFlu – were mostly one-word names, and of all drug names only 17.97% were phrases. Overall, 52.26% of commands were phrases, containing two to five words. The MEDIC speech corpus is used as an example to show that this methodology can also be used to recognize phrases (88).

The LIEPA project – Services Controlled by Lithuanian Speech – ended in August 2015, and produced the LIEPA speech corpus. The LIEPA Lithuanian Speech corpus is a phonetically representative database of Lithuanian spoken words adapted for scientific research, the development of speech technology, and the provision of electronic services.

The LIEPA corpus consists of two parts: Part 1 – a part of the speech corpus designed for speech recognition purposes; and Part 2 – a part of the speech corpus designed for speech synthesis purposes. The speech corpus is composed of 100 hours of speech data for Part 1, and 13 hours of speech data for Part 2. The quantity of texts for Part 1 is 78, 33 of which cover words and phrases, and 45 of which cover continuous speech. Speakers had to read 5–6 texts, and the lists of words and phrases mostly involved exact commands required by the speech recognition research group. The texts of continuous speech spanned descriptions of UNESCO objects, protected animals, and food. There were 376 speakers for Part 1: 116 speakers from schools and 260 from the main site (university students and invited speakers). Of these, 248 were female and 128 male. Four speakers were selected for Part 2 (156).

The basic phoneme set included 92 phonemes: long and short vowels, soft and hard consonants, diphthongs (vowel-vowel), and affricates with accent information later obtained. This phoneme set reflects the main attributes of the Lithuanian language and includes accent information, which is rarely obtained without a tool specific to the language.

The part of the LIEPA speech corpus selected for the investigation of isolated commands contains ten digits names from 0 to 9 uttered by 50 speakers (41 women, 9 men). The exclusive feature of this corpus is that it contains only one utterance of each digit by each speaker.

Another part of the LIEPA speech corpus used for phrase recognition was selected from Part 1. This part of the corpus consists of 10 separate sets, marked: Z000, Z001, Z020, Z021, Z022, Z023, Z024, Z060, Z061, and Z062. The Z060 part of the corpus was selected for testing because it contains many speakers, and more than one third of the corpus is composed of phrases. It contains 143 speakers (108

women, 35 men) and 26 commands (18 phrases and 8 isolated words). The individual specifications of all 10 parts are presented in Table 3.2.

Table 3.2. Speech corpus LIEPA Part 1 specifications

Corpus part no.	Speakers	Isolated commands	Phrases	Overall commands
Z000	138	56	0	56
Z001	146	30	0	30
Z020	142	16	15	31
Z021	32	21	46	67
Z022	32	13	48	61
Z023	32	9	37	46
Z024	32	9	38	47
Z060	143	8	18	26
Z061	30	16	3	19
Z062	29	280	1	281

For the evaluation of the hybrid approach to connecting recognizers in real conditions, we chose a simpler method: adding white noise at an SNR of 5 dB to the NAMES3 speech corpus, and using this noisy corpus in the experiment connecting two recognizers. The freely distributed Sound eXchange (SoX) program was used for adding noise to the speech corpus.

SoX is a cross-platform (Windows, Linux, MacOS X, etc.) command line utility that can convert various formats of computer audio files into other formats. It can also apply various effects to these sound files and, as an added bonus, can play and record audio files on most platforms. The synth command can be used to generate fixed or swept frequency audio tones with various wave shapes, or to generate wide-band noise of various "colors." An example command line for adding white noise to the audio file FOMEALB819.wav follows:

sox.exe E:\Noise\FOMEALB819.wav noise.wav synth whitenoise vol 0.1 && sox -m E:\Noise\FOMEALB819.wav noise.wav D:\Noise\SNR5\FOMEALB819.wav

3.4. The creation of Lithuanian digit name transcriptions

The names of ten Lithuanian digits: "nulis," "vienas," "du," "trys," "keturi," "penki," "šeši," "septyni," "aštuoni," and "devyni," were chosen for experimentation with German, English, French, and Spanish language recognizers. Firstly, Lithuanian digit names were rewritten into transcriptions using "synthesis" – i.e., each Lithuanian digit name was synthesized with different language synthesizer using the English, German, French, and Spanish UPS Alphabet (150) to prepare the transcriptions of Lithuanian digits. The foreign transcriptions most similar to the Lithuanian pronunciation of digit names were selected for further testing. The number of transcriptions found for each digit was unequal: for a short digit, such as "du," 7

transcriptions were enough, but for longer digits (“septyni,” “aštuoni”) up to 10 transcriptions were selected. Spanish language digit transcriptions were selected using a synthesizer (Table 3.3). Other languages transcription variables are located in Annex 1.

Table 3.3. Spanish language digit transcription selected using a synthesizer

Digit	Transcriptions
0	Nulis; Nuhlihs; Nuhlis; Nulihs; Nulejs; Nuljs; Nuls; Nwlis; Nwljs; Nulys.
1	Vienas; Bjenas; Vihehnahs; Vihehnas; Vihenas; Viehnas; Bienas.
2	Du; Duh; Duw; Duuw; Duuh; Dw; Dwa; Dwu; Duu.
3	Trys; Tryis; Tris; Triys; Trris; Trriis; Triis; Tdxis; Tdxiis; Tdxjis; Tdxjjs; Trriiis.
4	Keturi; Keturri; Ketudxi; Keturih; Kehtuhrih; Keturii; Kewturi; Keaturi; Keturrii; Ketudxrri; Ketury; Kewturih; Keaturih; Keaturii.
5	Penki; Peanki; Pewnki; Penkih; Penkii; Penkiih; Peankii; Peankiih.
6	Sesi; Sheshi; Chechi; Scheschi; Shehschi; Sheschii; Sheaschii; Sheaschiih; Shechii; Sheashii; Sheshii; Seasese; Cheasii; Chechii.
7	Septyni; Septynii; Septyniih; Septini; Septinii; Septiniih; Seaptinii; Seaptiinii; Septiinii.
8	Astuoni; Ashtuoni; Achtuoni; Ashtuonii; Astuonii; Achtuonii; Ashtwonii; Ashtuonji; Ashtuhohnii; Ashtuhohniih.
9	Devyni; Deviinii; Deviiniih; Debini; Dewini; Deaviinii; Deaviiniih; Debjinji; Debjinese.

Figure 3.6 illustrates the algorithm for the selection of the initial set of digit transcriptions, for which a grammar was generated and uploaded in the *Default pronunciations* pronunciation editor (Fig. 3.5). For each digit and for each different language recognizer, selection tests were prepared (40 tests overall). This test was performed by one male and one female speaker. A single digit grammar example for the selection of transcriptions is shown in Figure 3.1.

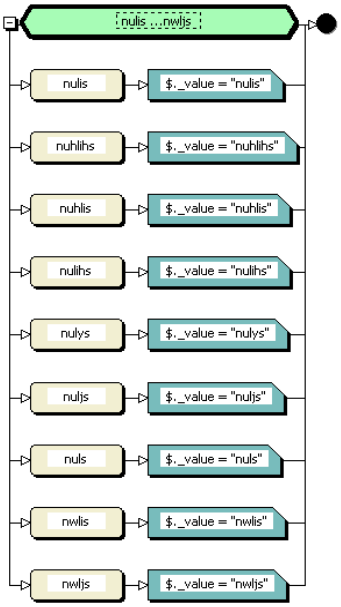


Figure 3.1. Single digit transcription selection test

Each digit was articulated 100 times through a microphone using MSS'2007 speech server. The most recognized transcriptions were called "winners," and used for the next step of transcription improvement by adding *Custom pronunciations* in the pronunciation editor (*PE*). Figure 3.4 presents the algorithm for the creation and selection of digit transcriptions.

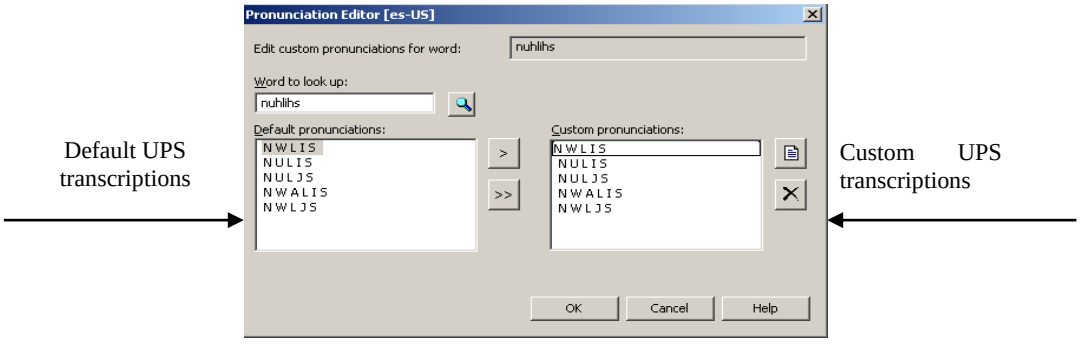


Figure 3.2. Pronunciation editor

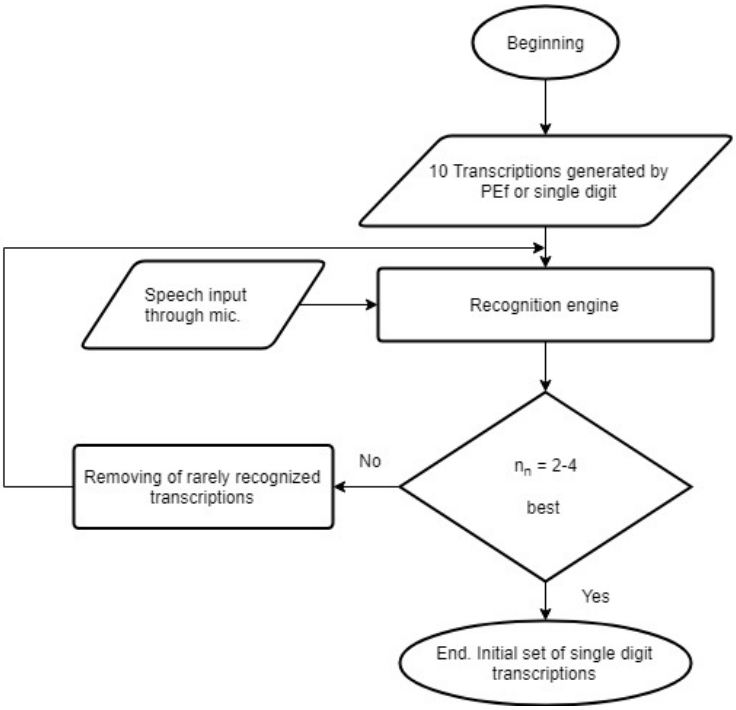


Figure 3.3. The algorithm for the selection of the initial set of digit transcriptions

As an example, we can see the most often recognized transcriptions for the digit "nulis":

- nuhlihs (for German language),
- nulis (for English language),
- nouluece (for French language),
- nuhlihs (for Spanish language).

If the same digit had more than one recognized transcription, all recognized transcriptions were used for further research.

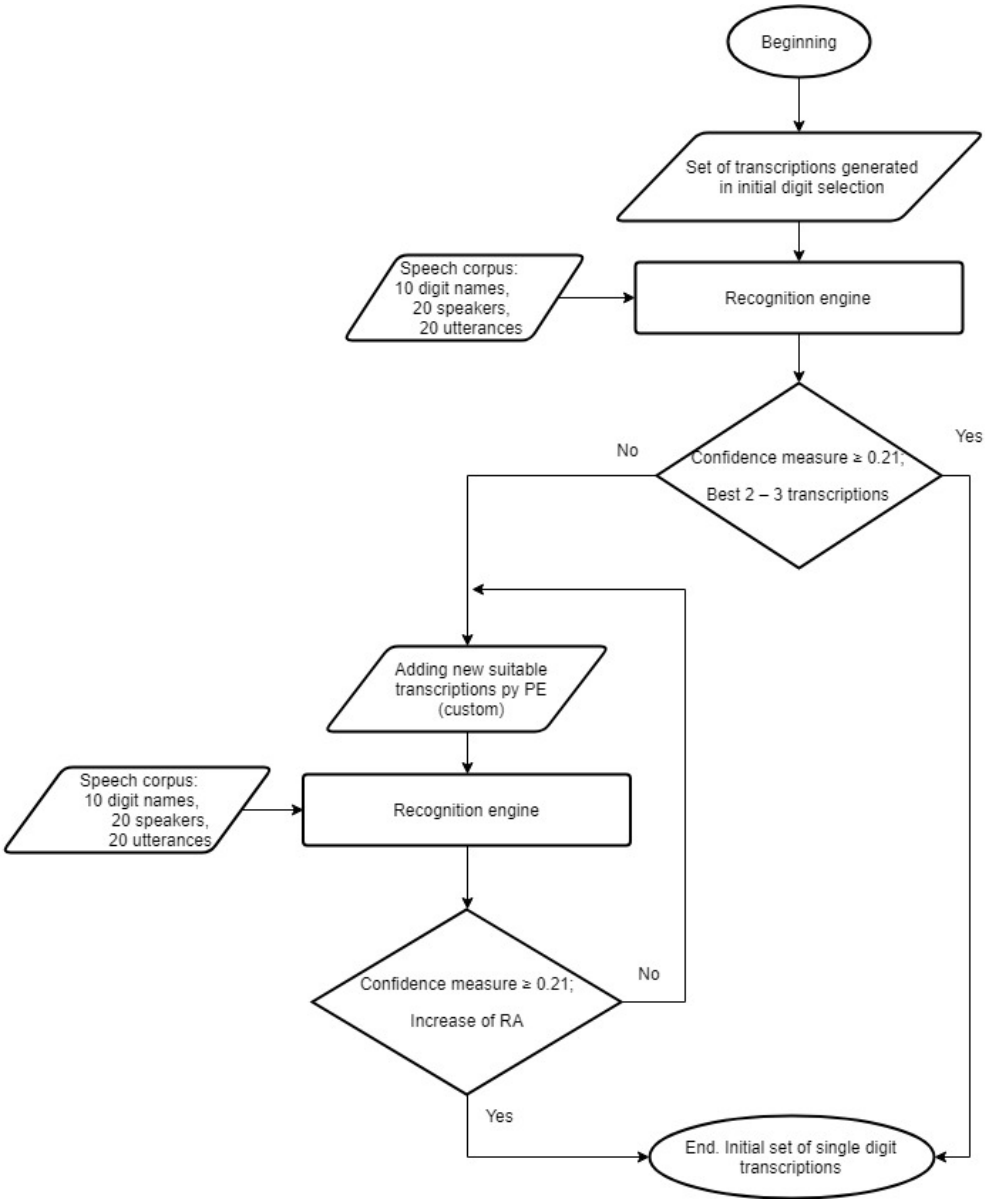


Figure 3.4. The algorithm for the creation and selection of digit transcriptions

3.5. The selection of names and words corresponding to Latin letters

It is obvious that the accuracy of letter recognition could be considerably improved by using the appropriate set of words equivalent to the Latin alphabet. The average recognition accuracy of the LETTERS speech corpus by the REC_SP recognizer was only 25.9%, indicating that spelt-out letters cannot be used for the recognition of disease codes. The accuracy of the NATO speech corpus was 67.2% (157). Therefore, an appropriate Lithuanian name was chosen for each letter:

“Antanas” fort the letter “a”; “Benediktas” for the letter “b”; and so on, using the NAMES1 speech corpus (a full list of names is presented in Annex 2). The selection of appropriate names was performed experimentally by testing the recognition accuracy of ten Lithuanian names for each letter and looking for the most often-recognized names. The algorithm of vocabulary selection for the name speech corpus is presented in Figure 3.5. Name selection was carried out using the adapted Spanish language recognizer included in the Windows 7 operating system.

For speech corpus vocabulary selection, iteration was carried out 3 times. Iterations were carried out in alphabetical order – starting with the letter “a” and full grammar (*PG*).

In each iteration step, recognition testing was executed with the full speech corpus of names, starting with the tested letter. Recognition results were calculated as follows: if recognition accuracy was 80% or more, then 1, 2, or 3 name transcriptions were selected as the best-recognized ones. In the recognition grammar, only the selected transcriptions of names were left, as other transcriptions were removed (*AG_R*). A list was created for each letter with tested names, with results presented in order of recognition accuracy.

If, in the second step, names with a recognition accuracy of 80% or more could not be found, all names beginning with the tested letter were left for the next iteration, except for obviously unrecognizable names (e.g., in the case of the letter “a” – *AG_B*).

In cases where the name recognition accuracy was below 80% and the recognition results showed that the name was mixed with a name that started with the other letters, it was permitted to remove the name from the grammar, unless it was the only remaining name starting with that letter.

In case of an emergency, it was permitted to remove only the last name starting with a different letter if it disrupted the recognition of the tested name. However, in such a case, another name previously withdrawn based on the recognition accuracy of the sequence list was returned to the grammar. This was only used in case of an emergency – when the removal of the disrupting name significantly improved the recognition accuracy of the tested name, and the disrupting name was the only name left for the selected letter.

Additional requirements were as follows:

- the final list should not conclude names with similar text fragments (*PG_X*): Daumantas, Eimantas; Mantas, Skirmantas; Gražvydas, Mažvydas; Aleksas, Feliksas, Iksas; Florijona, Jonas, Ulijona;
- at least one name should be assigned to each letter.

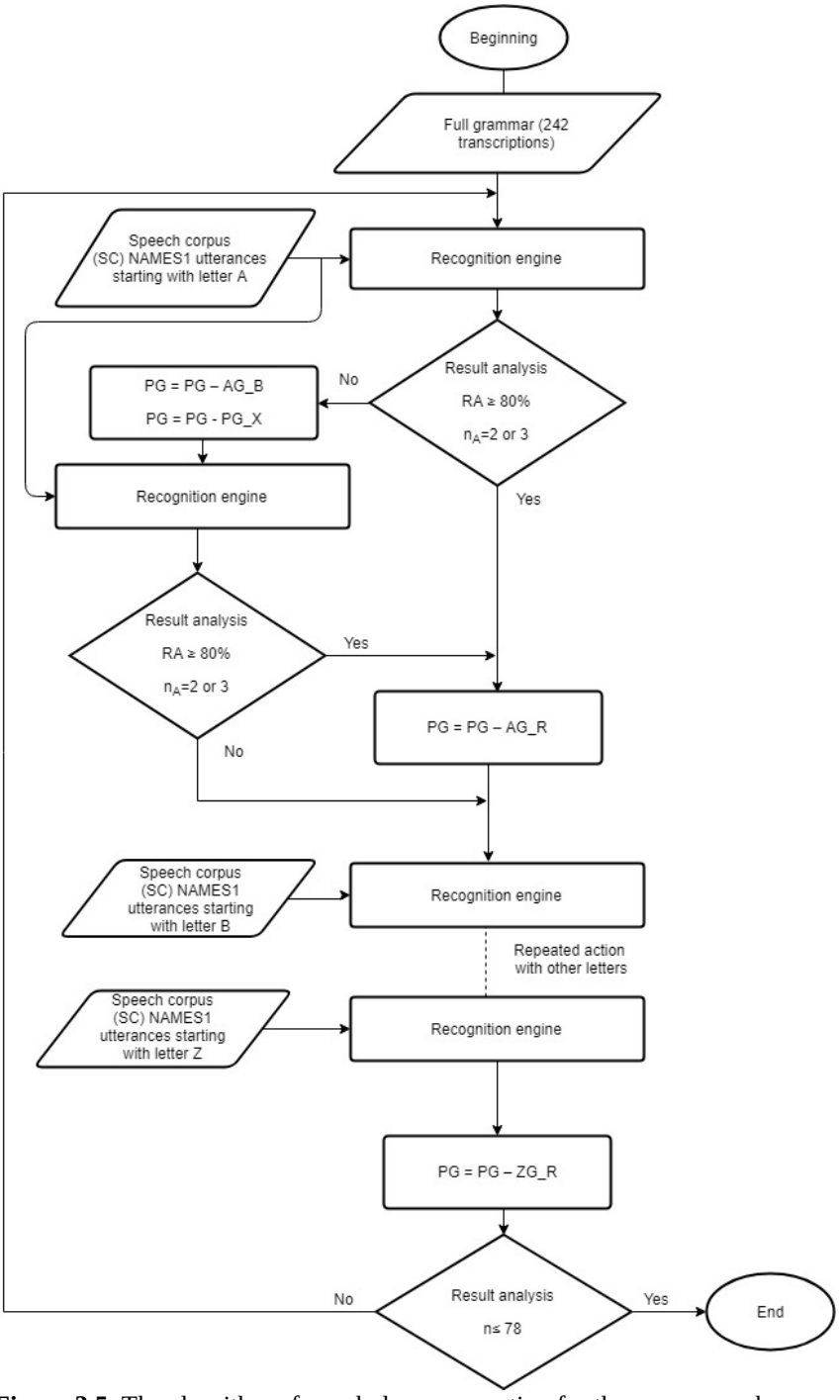


Figure 3.5. The algorithm of vocabulary preparation for the name speech corpus (one iteration)

After initial selection, the NAMES2 speech corpus was composed of 70 names. The next selection of the final set is presented in Figure 3.6. After this selection, the NAMES3 speech corpus was formed, representing the final set containing 26 names and words equivalent to Latin letters. In order to select the final set, the recognition accuracy of one name (*RA_L*) alone should be higher than 85%, whilst at the same time overall recognition accuracy (*RA*) should be higher than 95%.

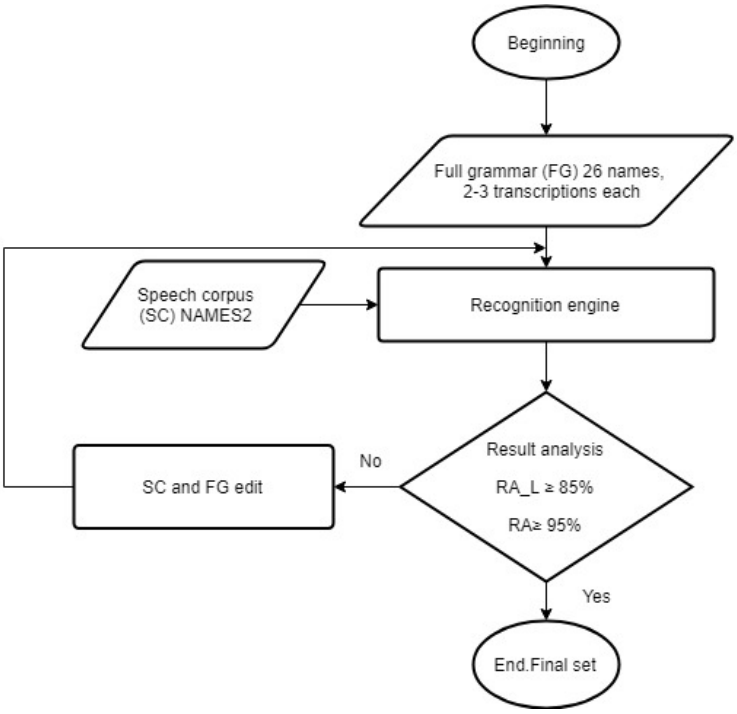


Figure 3.6. The selection algorithm of names equivalent to the 26 letters of the Latin alphabet.

3.6. Isolated word command recognition using HTK

HMM technology was used for the creation of the Lithuanian REC_LTp and REC_LTw recognizers. For the creation of the acoustic models of the Lithuanian recognizers, the HTK v.3.2 open code software toolkit was used (112). The algorithms of phoneme-based HMM and word-based HMM recognizers are presented in Figures 3.7 and 3.8.

The tools in the HTK framework (112) are designed to perform different tasks in building the HMM. Building a speech recognition system on HTK requires tools that can implement four stages: data preparation, training, testing, and results analysis, as shown in Figure 3.7 for phoneme-based and Figure 3.8 for word-based HMM.

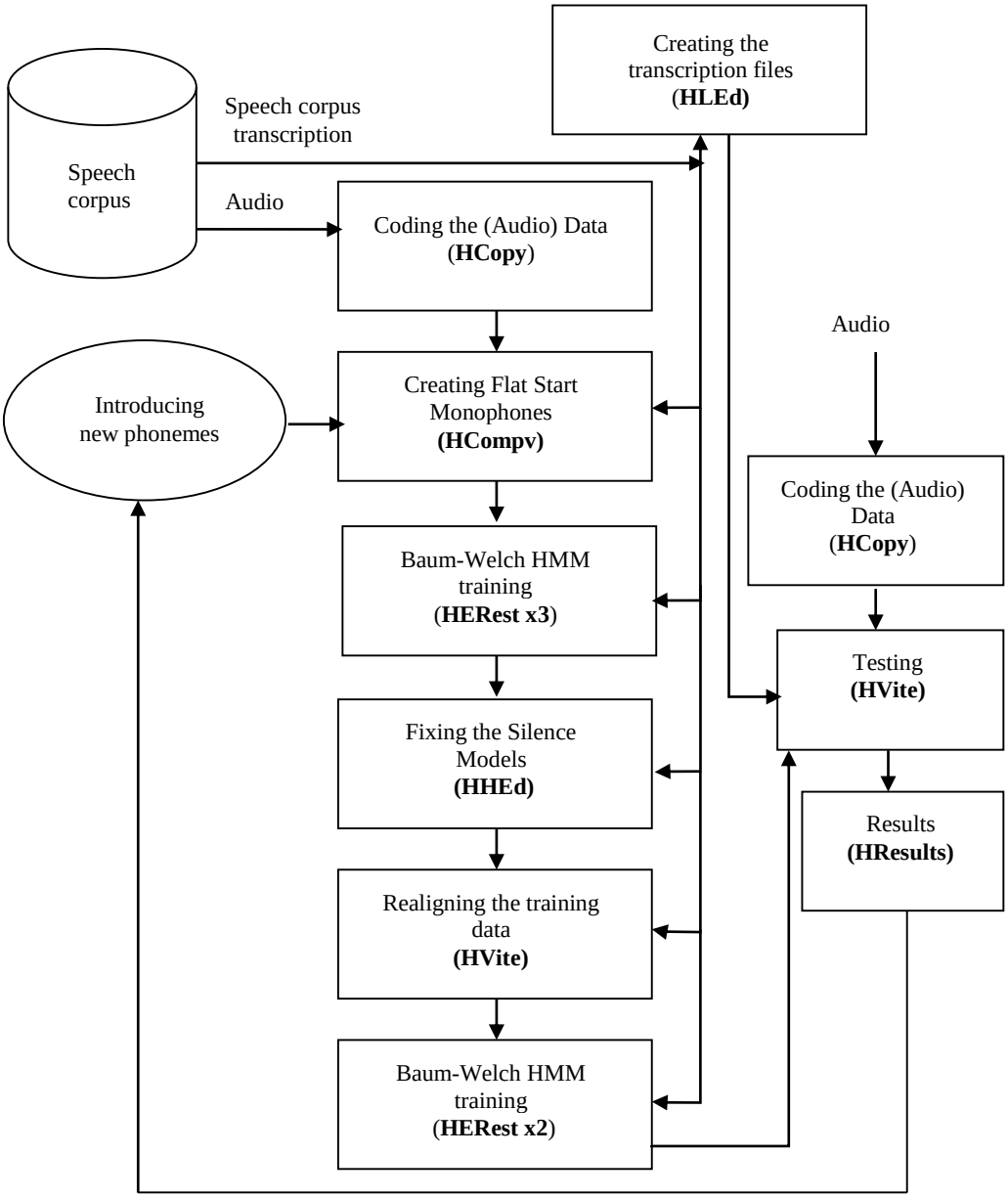


Figure 3.7. Stages of building a phoneme-based speech recognizer with HTK

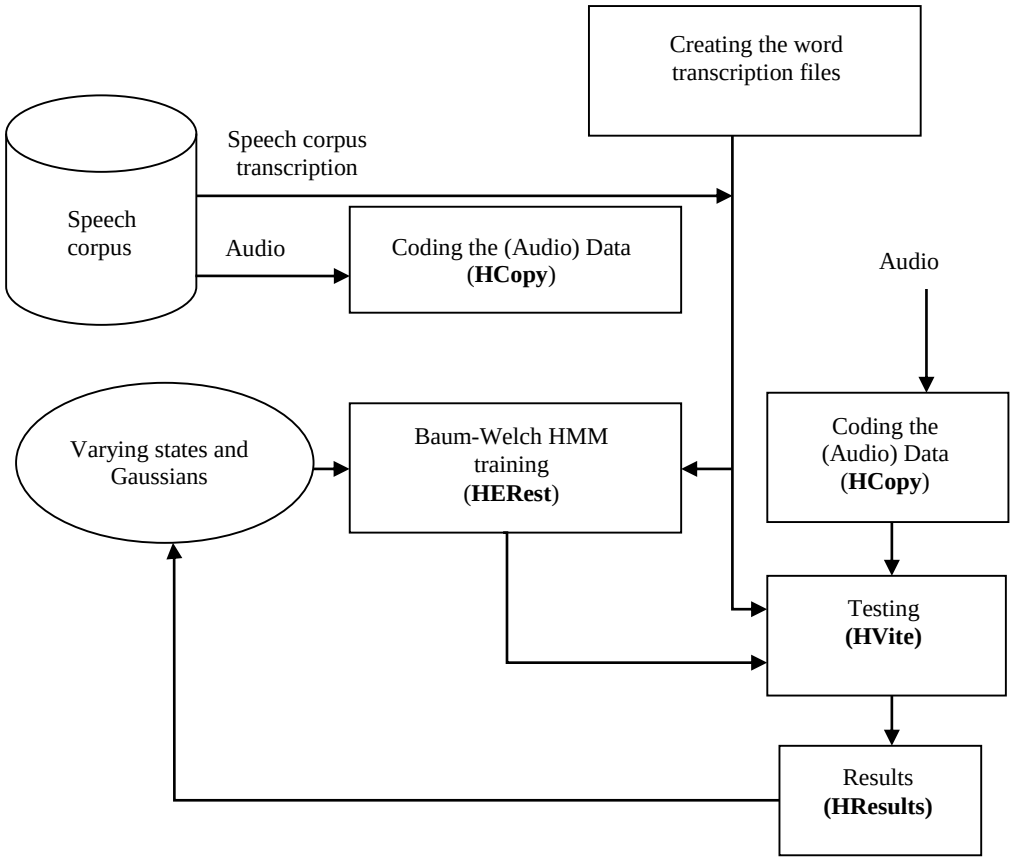


Figure 3.8. Stages of building a word-based speech recognizer with HTK

In the data preparation stage, the HCopy tool processes the voice signals obtained from the microphone or speech corpus into codebooks (MFCC) according to the feature extraction method of a speech recognition system. HLEd produces transcriptions to read a list of HMMs and a set of voice accents. Next, at the training stage, HTK provides tools to estimate the parameters for the HMM – HcompV and HRest – of which HcompV is used to initialize the values of the parameters. HcompV calculates the expectation and variance of each Gaussian component in the HMM definition to make them almost equal to the expectation and variance of the speech training data. HRest estimates the parameters of a data segment using the Baum-Welch algorithm. In phoneme-based HMMs, new phonemes are introduced in training; and in word-based HMMs, the number of states and Gaussian mixtures are varied in order to obtain higher recognition accuracy. HHEd is used to create a new HMM after parameter adjustment. The result is the creation of HMMs in accordance with the dictionary of words to be recognized. There are a number of supportive tools in HTK: HParse is used to change a syntax file of the dictionary into a semantic network and provides the possibility to arrange the words in order; HCopy is used to extract features to identify a word; and the HVite tool applies the Viterbi algorithm

for speech recognition based on the constraints of the the HMM model, the dictionary, and the grammar structure.

3.7. Isolated command recognition using two different recognizers and a noisy speech corpus

Using the Kaldi package, the acoustic data includes: gender information on the speakers (spk2gender), the identifier, and the audio data for each utterance (wav.scp); the transcripts for each utterance (text); the mapping between utterances and speakers (utt2spk); and the corpus's transcript (corpus.text) (57). The language data includes the lexicon (the list of words and phrases together with transcriptions), and both the non-silence (the set of Lithuanian phonemes) and silence phone information. The list of all words and phrases of the speech corpus was used as a language model. Scripts and tools for experiments were used from speech recognition examples, using Kaldi and the Wall Street Journal speech corpus presented in *kaldi/egs/wsj/s5*.

A CTC-based approach was selected for the connection of two different recognition engines for two main reasons:

- Without using a language model, attention models outperform CTC models trained on the same corpus, but it was found that CTC models are significantly more stable, easier to train, and ensure better recognition results if a language model is used (the performance of both models against the Hub5'00 benchmark is presented by Battenberg et al. (70);
- Though very good recognition results were achieved for part of the LIEPA speech corpus using an attention-based model (99), the preparation of data for the model and the implementation of the CTC model is simpler compared to the attention-based model.

The Deep Speech 2 model (69) was selected as the CTC-based recognition model, implemented using the TensorFlow package. TensorFlow is a powerful data flow-oriented machine learning library created by Google's Brain Team, and made open source in 2015. It was designed to be easy to use and widely applicable to both numerical and neural network-oriented problems, as well as to other domains.

Deep Speech 2 is an end-to-end DNN for ASR based on a Baidu engine (69). It consists of two convolutional layers, five bidirectional RNN layers, and a fully connected layer. The feature in use was a linear spectrogram extracted from audio input. The network uses CTC as the loss function.

Preparing data for the Deep Speech 2 model is very simple. Two files should be prepared:

- the name and location of the audio file, the size of the audio file, and the transcription of the word or phrase of the audio file; three separate files should be prepared for training, evaluation and testing;
- a vocabulary file consisting of the list of phonemes.

Due to the small size of the NAMES3 speech corpus, some Deep Speech 2 parameters were reduced:

- *rnn_hidden_size=256*
- *rnn_hidden_layers=3*

- *stride_ms=10*
- *window_ms=20*
- *batch_size=12*
- *train_epochs=10*

The size of the audio files tested was 1/7th of the whole speech corpus, as 7-time cross-validation was used. For the evaluation set, 30% of the training audio files were used.

3.8. Metrics

The performance of recognition systems was measured by recognition accuracy (RA), defined as:

$$RA = \frac{R}{n} \times 100\%, \tag{65}$$

where *n* is the number of words used in the test, and *R* is the number of correctly recognized words.

A confidence measure was calculated following two patterns: one for the results obtained by the cross-validation principle; and another for the results obtained by testing one data set. That is to say, calculation of confidence intervals by normal distribution, when variance is unknown, and calculation of confidence intervals by approximating the binomial distribution to the normal distribution.

Confidence intervals were calculated with 95% confidence. Intervals were calculated following two patterns: one for the results obtained by the cross-validation principle; and another for the results obtained by testing one data set. Put simply: calculation of confidence intervals by normal distribution, when variance is unknown; and calculation of confidence intervals by approximating the binomial distribution to normal distribution (158).

To identify *n* different sets, the formula for finding the normal distribution parameters with unknown variance in the confidence interval was used. According to this formula, the accuracy of the estimate of the mathematical hope is the product of the ratio of the quantile $t\alpha/_{2;n-1}$ and the unshifted value *s* of the mathematical hope to the square root of the sample size *n* (*n* is the mean of the recognition results RA).

$$\varepsilon = t\alpha/_{2;n-1}\frac{s}{\sqrt{n}}, \tag{66}$$

Then,

$$s = \sqrt{\frac{\sum_{i=1}^{N}(l_i - l_{av})^2}{n-1}}, \tag{67}$$

where l_i represents the *i*-th measurement value, and l_{av} the average of all measured values. Quantiles $t\alpha/_{2;n-1}$, when α-0.05, are found in Student's *t*-distribution tables.

To identify one test set, the formula for finding the confidence intervals of the approximation of the normal distribution by binomial distribution was used. According to this formula, the accuracy of the *ZT* estimate is the product of the argument for which the value of the standard normal distribution *N* (0,1) is equal to

the given confidence probability, and the ratio of the standard deviation *s* to the square root of the sample size *n*.

$$\varepsilon = z_{1-\alpha}\frac{s}{\sqrt{n}}\ , \tag{68}$$

Then,

$$s = \sqrt{RA(1-RA)} \tag{69}$$

The Wilcoxon test is a nonparametric test designed to evaluate the difference between two treatments or conditions where samples are correlated. This test was used to evaluate the connection of two recognizers.

The Kruskal–Wallis test is a non-parametric alternative to the one-factor ANOVA test for independent measures. It relies on the rank-ordering of data rather than calculations involving means and variances, and allows for the evaluation of the differences between three or more independent samples (treatments). This test was used to evaluate the differences in connecting three recognizers.

3.9. Data mining software and classifiers

For the connection of several recognizers, suitable software for data mining can be chosen from over 600 commercial and open-source systems (158). In the period from 2008–2010, a study was carried out during which the users of data mining systems indicated which systems they use in ongoing projects (159). It was evident that the most commonly used open-source systems are: RapidMiner (160), R (161), KNIME (162), Weka (163), and Orange (164). Meanwhile, the often-mentioned Excel and MATLAB are used only supplementary – they are commonly used together with the more popular previously mentioned open-source data mining systems (159).

In one detailed overview of six open-source data mining systems, it is stated that there is no "best" data mining system, but a choice is offered between four data examination packets: RapidMiner, R, Weka, and KNIME (165). Similar results have been obtained in other research (166) in which 12 data mining systems were analyzed, and YALE (an older version of RapidMiner), KNIME, AlphaMiner, Weka, and Orange received the most positive reviews. Nine types of classification objects and six types of classifiers were examined in another work (167), where Weka was evaluated very favorably. Based on this review, for studies connecting recognizers, the Weka packet was selected (163). It is also one of the most widely used pieces of open-source data mining software in Lithuania (14).

In Weka, several dozen classifiers are introduced, and so it was necessary to choose the most efficient classification methods from them. This initial selection was based on a review of the literature (168). The most popular are Naive Bayes (NB), K-Nearest Neighbor (kNN), decision tree, Multilayer Perceptron (MP), and support vector machine (SVM). The most popular classifiers of the decision tree type are the C4.5 and random forest (RF) classifiers. Other works (166) have examined the OneR and ZeroR classifiers alongside the previously mentioned NB, C4.5, SVM, and k-NN classifiers. When choosing a classifier, Demšar, Curk, and Erjavec's (164) overview of the 10 the most popular data mining algorithms – including C4.5, NB, k-NN, SVM,

AdaBoost, CART (Classification and Regression Trees) classification algorithms, and some algorithms for clustering – can be referred to. If one is to refer to Jovic, Borkic, and Bogunovic's overview (165) and other work by the same authors (169) – together with the already mentioned C4.5, RF, and NB classifiers – then we should also examine the RIPPER classification algorithm. Weka has no regression CART algorithm; instead an MLR (Multinomial Logistic Regression) algorithm and a ZeroR classifier was selected over two similar OneR and ZeroR classifiers.

Random forests are one of the most successful machine learning models for classification and regression. Random forests are ensembles of decision trees. They combine many decision trees in order to reduce the risk of over fitting (170). Like decision trees, random forests handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearities and feature interactions. Random forests train a set of decision trees separately, and so the training can be done in parallel. The algorithm injects randomness into the training process so that each decision tree is slightly different. Combining the predictions from each tree reduces the variance of the predictions, improving performance on test data.

3.10. The technique of connecting recognizers

A hybrid recognizer has the potential to exploit the advantages of several recognizers at once. An example structure of a hybrid recognizer is given in Figure 3.9. In this example, a hybrid recognizer is comprised of the REC_SP (adapted Spanish language) recognizer, having an integrated speech boundaries detection unit, and the REC_LTw (Lithuanian) recognizer and decision-making unit, which realizes the hybrid decision-making rule(s).

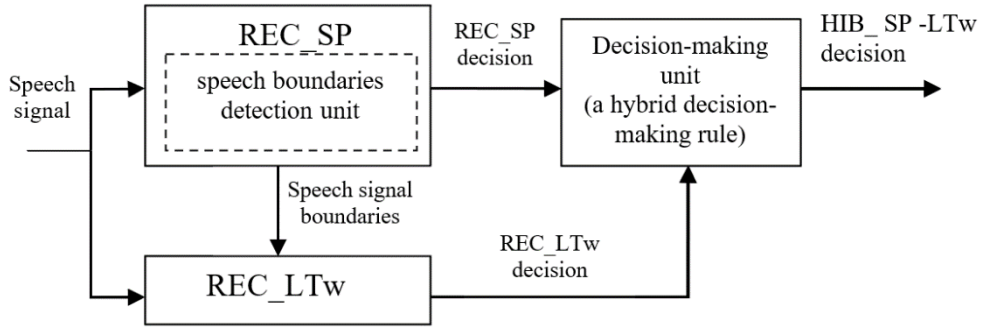


Figure 3.9. Hybrid recognizer REC_SP/REC_LTw structure

The speech signal is primarily directed to the REC_SP recognizer, which determines the boundaries of the command and provides a fragment of the signal to the REC_LTw recognizer. The REC_LTw recognizer also provides a decision to the previously mentioned unit. If the decisions differ, the unit has to decide which of the two decisions should be submitted for the user to see as the final answer.

The parallel use of two recognizers is useful, as when one recognizer provides the wrong answer, the other can make the correct decision and vice versa. Non-

Lithuanian recognizers have advantages that include signal detection, noise processing, and the availability of other modules.

The recordings were segmented into subsets based on the recognition decisions provided by recognizers. These subsets reveal, in detail, evidence that the REC_LTw and REC_SP recognizers added value to one another – this is described in more detail in Table 5.2.

The task of separating two subsets was formulated, and resolved by using the names TF (where the decisions provided by the devices do not match, and REC_LTw made the correct decision), or FT (where the decisions provided by the devices do not match, and REC_SP made the correct decision). The most important element in a hybrid recognizer is a hybrid decision-making unit, made by adapting machine learning. The answers used for this learning were chosen specifically when the decisions from the recognizers differed. Every object involved in the learning process was gathered from the decisions of both recognizers for a specific recording (both TF and FT class decisions).

The benefit of a hybrid recognizer was calculated by using the simple "blind" decision rule, which states that "if the decisions of the devices differ, choose the one made by the better recognizer" – thereby providing TF/(TF+FT)*100% accuracy. The study showed that a result obtained by a new hybrid decision rule was useful only when it achieved a higher score than this "blind" rule score.

Every single study object was defined by attributes, which are explained in Table 3.4. The main attributes given in Table 3.4 are "lt_prob" and "sp_prob" – as an example, "sp_prob" represents the the measure of confidence in the decision of REC_SP, and the closer this number is to 1, the larger the possibility that the recognizer has made a correct decision. The attributes "lt_delta_prob" and "sp_supp" were created based on the fact that the REC_LTw recognizer provides from 1 to 3 answers, presenting them in descending order of priority. If the first decision was correct, the phrase was considered to have been recognized correctly.

Table 3.4. Description of the features used for the combination of two recognizers

Feature	Description
sp_prob	The measure of confidence in the decision of the REC_SP recognizer
sp_supp	The difference between the average logarithmic probabilities of the first, second, or third alternatives provided by the REC_LTw recognizer in cases when REC_SP's decision coincides with REC_LTw's second or third alternative. If the REC_LTw recognizer does not provide an alternative decision, this attribute takes a value of 10
lt_prob	The average logarithmic probability of the REC_LTw recognizer's decision
lt_delta_prob	The difference between the average logarithmic probabilities of the first and second alternatives provided by the REC_LTw recognizer. If the recognizer does not provide an alternative decision, this attribute takes a value of 10
gender	The speaker's gender (m, f)
lt_a, …, lt_ž	The proportion of the number of certain letters to the number of all letters in the REC_LTw recognizer's decision, % (for example, if the decision is "du," then lt_u=lt_d=50%).
sp_a,..., sp_ž	The proportion of the number of certain letters to the number of all letters in the REC_SP recognizer's decision, %

The number of other features was determined by the number of letters in the Lithuanian language. The goal was to create as many attributes as possible, and their influence was investigated in further research. The effect of these attributes is investigated through a process whereby certain attributes are removed from files with the help of the WEKA analysis system, which automatically provides a calculated accuracy. This investigation was performed using the 10-times cross-validation method, with 90% of objects involved in learning and the left-over 10% in testing.

3.11. The technique of classification with the WEKA package

After a thorough literature analysis (section 3.8), the WEKA data analysis system was chosen for the purpose of achieving the best possible results in the connection of two recognizers. This data analysis system has many classifiers, of which we were required to choose the most effective. Classifier selection was comprised of 10 candidates: kNN, RIPPER, NB, RF, C4.5, ZeroR, SVM, AdaBoost, MP, and MLR (171).

For the connection of recognizers, two different methods were applied:

1. Ordinary 10-times cross-validation with the graphical WEKA interface. One file with the attributes of all speakers was prepared, and then by default WEKA randomly distributed the data: 90% for training, 10% for testing. It performed the classification 10 times, changing the set of test objects, and then calculated the average of the obtained results. This classification method allowed for the prediction of the accuracy of the classification (and at the same time, the accuracy of the hybrid recognizer) for the "known speaker" (one of the speakers of the speech corpus).

2. The more complex *n*-times cross-validation method, with *n* number of speakers. Here, *2*n* files were prepared: for training, the features of *n-1* speakers are taken; and for testing, the features of 1 (unused) announcer. The classification was carried out *n* times through the command line, giving a file with the attributes of *n-1* speakers for training, and a file with the attributes of the unused announcer for testing. This was repeated *n* times by changing the unused announcer, and the results were then averaged manually. The results of such a classification allowed for the prediction of the classification accuracy (at the same time, the accuracy of the hybrid recognizer) for an "unknown speaker." Due to the high volume of calculations, instead of *n* times cross-validation, *n/2*, *n/3* (and so forth) cross verifications were carried out – as such the results were less accurate.

The effectiveness of the hybrid decision-making rule, using the *n*-times cross-validation method, was calculated using the "SimpleCLI" system tool in the WEKA data analysis system. In the command line of this tool, data on the classification type, classifier, classification training data directory, and test data directory were specified.

The "learn.arff" and "test.arff" files were used for training and testing, as these files contain features of the tested data. Research was then carried out with all of the classifiers chosen for the test and with all prepared data files of the speakers, specifying different speaker and classifier data files each time.

The analysis of hybrid decision-making performance using the 10-times cross-validation method was conducted using the "Explorer" system tool in the WEKA data analysis system.

3.12. Chapter summary

1. A technique for the selection of names and words which are appropriate for the recognition of Latin letters was created. This was based on the creation of a speech corpus including a large number of names (up to 10 names per letter), and the iterative selection of the most recognizable names/words with the adapted language recognizer.
2. A technique was prepared for the selection of isolated word command transcriptions using a voice server lexicon editor, a recognizer for non-native language, and UPS, verbal, or mixed transcriptions.
3. Taking into account the fact that the annotation of an examined speech corpus requires a lot of time and human outlay, studies on the recognition of a speech corpus with a HTK packet are limited to word-based and phoneme-based HMMs, and HMMs of contextual phonemes are only attempted.
4. A technique was proposed for the recognition of isolated commands by selecting the number of HMM states and Gaussian mixtures in a word-based HMM, or introducing new monophones to a phoneme-based HMM.
5. A method was proposed for the connection of several recognizers, using machine learning and selecting the most effective classifier with the WEKA packet.

4. RECOGNITION RESEARCH

4.1. Research into an adapted language recognizer for Lithuanian voice commands

4.1.1. The recognition of Lithuanian digit names using Microsoft Speech Server

Four tests were prepared for the names of ten Lithuanian digits (0–9) with different grammars for each language recognizer, using UPS and Speech Grammar Editor (SGE). The following language recognizers were used: German – Microsoft Speech Recognizer 9.0 for MSS (German-Germany); English – Microsoft Speech Recognizer 9.0 for MSS (English-US); French – Microsoft Speech Recognizer 9.0 for MSS (French-Canada); and Spanish – Microsoft Speech Recognizer 9.0 for MSS (Spanish-US). All of the "winning" transcriptions were used in these grammars. Each digit was pronounced into a microphone 100 times each by a male and female speaker before being exposed to a recognizer. The results then were gathered, analyzed, and processed.

The RA of two different speakers using adapted language recognizers is presented in Table 4.1, and the average confidence measure in Table 4.2.

Table 4.1. Lithuanian digit names RA with four adapted language recognizers

Speaker	Speech server implemented recognizer RA, %			
	German	English	French	Spanish
KR, male	58.4	76.4	52.8	88.2
GB, female	51.8	59.0	76.8	98.8
Average	55.1	67.7	64.8	93.5

The strongest RA results for the 10 Lithuanian digit names were achieved with the Spanish recognizer, which produced an average RA for the male speaker of 88.2%, and 98.8% for the female speaker. Overall, the average RA of the Spanish recognizer was 93.5%.

Table 4.2. Confidence measure of Lithuanian digit recognition for four adapted language recognizers

Speaker	Speech server implemented recognizer confidence measure			
	German	English	French	Spanish
KR, male	0.44	0.48	0.48	0.60
GB, female	0.42	0.37	0.57	0.77
Average	0.43	0.43	0.53	0.68

The results in Table 4.2 show that the strongest confidence measure was also attained using the Spanish language recognizer: 0.6 for the male speaker, and 0.77 for the female speaker (the confidence measure may vary from 0 to 1).

For the next stage of the research, the SKAIC30 speech corpus was used. Since speech server requires speech input in a telephony format, this speech corpus was adapted by down-sampling the speech corpus from the original 16 kHz to an 8 kHz sampling rate. The previous research results show that the Spanish recognizer was the most accurate of the four in MSS'2007. It was therefore selected for recognition tests using this speech corpus.

Grammar was prepared using 36 transcriptions. The visual display of the grammar window in MSS is presented in Figure 4.1.

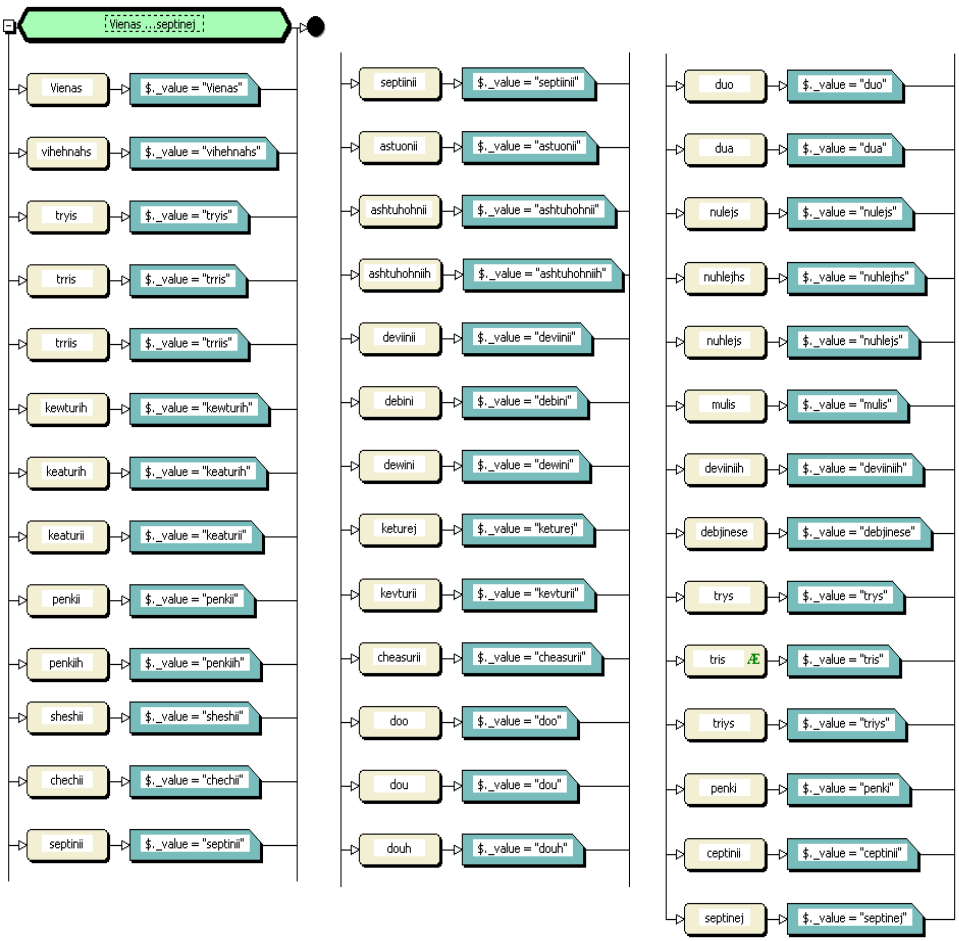


Figure 4.1. A visual display of voice command recognition grammar

The average RA of ten Lithuanian digit names using REC_MSS is presented in Table 4.3. The Spanish speech engine enabled the achievement of an overall RA of 99.12±0.88% for ten Lithuanian digit names. This is a significantly higher RA result

for the same digits than the 92.5% achieved in the thesis of Maskeliunas (83 p. 111), where a speech corpus of 10 speakers, 10 digits, and 20 pronouncements of each digit was used with an adapted language recognizer.

Table 4.3. Average RA and confidence measure of ten Lithuanian digit names with REC_MSS using the SKAIC30 speech corpus.

Command	Recognition accuracy, %	Confidence measure
NULIS	99.67±0.22	0.87
VIENAS	100	0.93
DU	98.83±0.42	0.90
TRYS	99.5±0.27	0.87
KETURI	95.5±0.80	0.74
PENKI	100	0.92
SESI	97.83±0.57	0.84
SEPTYNI	99.83±0.16	0.90
ASTUONI	100	0.86
DEVYNI	100	0.87
Average RA % with 95% confidence level	**99.12±0.88**	**0.84**

As can be seen throughout, a strong confidence measure was achieved for all digits. However, the lowest confident measure – a still-modest 0.74 – was obtained for the digit "four." The highest confidence measure was achieved for the digits "one" and "five" – 0.93 and 0.91, respectively. The total average confidence measure for all ten digits was 0.87.

4.1.2. Lithuanian digit name recognition using the Spanish recognizer

Earlier experiments revealed that Microsoft Speech Recognizer 8.0 (Spanish-US) provides significantly better results for Lithuanian digit name recognition compared to the other recognizers implemented in the Windows 7 operating system. Therefore, the abovementioned recognizer was selected as the adapted foreign language recognizer.

The structure of recognition grammar enables the use of multiple UPS-based or word-based transcriptions of commands and synonyms of commands together, according to the SRGS grammar specification. As such, UPS-based transcriptions prepared using the pronunciation editor in MSS were used by the REC_SP recognizer in three forms:

- UPS transcription in MSS;
- word-based transcription, obtained from removing spaces between phonemes in the UPS transcription;

- mixed transcription, obtained by mixing the grammar of the UPS and word-based transcriptions.

Testing experiments using the SKAIC30 speech corpus were carried out with the Spanish recognizer 8.0 (Spanish-US), using the same Lithuanian digit name transcriptions as in the MSS'2007 speech server research. The average recognition accuracy of ten Lithuanian digit names with different profiles is shown in Table 4.4.

Table 4.4. RA of digit names by Spanish recognizer 8.0, with different profiles

Command	Profile grammar RA, %						
	Default, word-based	Default, UPS	Female, word-based	Female, UPS	Male, word-based	Male, UPS	Male, Mixed transcriptions
NULIS	42.33	61.00	55.00	54.50	70.67	75.67	80.67±1.53
VIENAS	91.17	93.67	93.33	93.33	96.67	96.33	95.83±0.78
DU	64.33	67.00	51.67	51.33	73.50	80.00	79.67±1.56
TRYS	98.00	98.17	96.50	96.50	99.00	99.17	99.00±0.39
KETURI	53.83	57.83	49.50	48.83	85.83	74.50	86.50±1.33
PENKI	97.17	95.67	90.67	90.83	98.17	98.67	98.67±0.44
SESI	97.33	100	98.83	98.83	100	100	100
SEPTYNI	97.67	98.00	96.17	96.00	99.17	99.50	99.17±0.35
ASTUONI	95.33	95.67	86.50	87.17	99.67	99.67	99.67±0.22
DEVYNI	75.50	78.00	80.67	80.67	72.00	86.50	81.33±1.51
Average RA % with 95% confidence level	**81.26 ±12.99**	**84.81 ±10.60**	**79.89 ±12.36**	**79.79 ±12.52**	**89.48 ±7.88**	**91.01 ±6.63**	**92.05 ±5.48**

Three different profiles were used for this research: the default profile, which is initially implemented in the Windows OS, and is not trained; and female and male profiles, which were trained by a female and a male speaker, respectively. Before first using Windows Speech Recognition, a microphone was set up. Speech Recognition uses a unique voice profile to recognize voices and spoken commands. Windows comes with a speech training tutorial to help teach the profile used for Speech Recognition. The tutorial takes approximately 30 minutes to complete.

With different profiles, two types of transcriptions are used: word-based and UPS. The results in Table 4.5 show that the male profile had better recognition results with both word-based and UPS transcriptions than the default or female profiles. A decision was made to conjunct the word-based and UPS transcriptions (the type of transcription used for a mixed set is highlighted) into one grammar set. After conjunction, recognition accuracy results increased to 92.05±5.48%.

4.1.3. Name and word recognition using the Spanish recognizer

The NAMES3 speech corpus consists of utterances of 22 Lithuanian names and four words (kju, Wasington, iksas, and ygrekas) which represent the basic Latin alphabet, consisting of the 26 letters used in disease codes. This corpus was used for name and word recognition research with Microsoft Speech Recognizer 8.0 (Spanish-US), which was chosen due to earlier research on Lithuanian digit names (section 4.1.2). Two profiles were trained on Windows OS: male and female, and a third default profile was untrained. Three grammars were prepared: word-based, UPS, and mixed (using both words-based and UPS transcriptions). The RA results are presented in Table 4.5.

Table 4.5. RA of names and words using REC_SP with different profiles

Command	Profile grammar RA, %				
	Female, word-based transcriptions	Default, word-based transcriptions	Male, word-based transcriptions	Male, UPS transcriptions	Male, Mixed transcriptions
Austėja	90.8	98.7	99.6	99.2	99.6±0.29
Boleslovas	96.0	97.7	98.9	98.3	98.9±0.48
Cecilija	96.3	97.1	99.8	99.3	99.8±0.20
Donatas	98.3	98.7	99.8	99.8	99.8±0.20
Eimantas	98.9	98.9	97.9	97.9	97.9±0.66
Faustas	98.7	98.9	98.3	98.7	98.3±0.60
Gražvydas	95.0	98.9	99.4	98.7	99.4±0.36
Hansas	98.9	98.9	99.0	100.0	99.0±0.46
Izaokas	99.4	98.5	98.3	99.2	98.1±0.63
Jonas	94.0	96.3	97.1	96	97.1±0.78
Karolis	100.0	100.0	97.9	97.7	97.9±0.66
Laima	97.9	99.4	98.9	99	98.9±0.48
Martynas	99.6	98.3	97.7	99.8	97.1±0.78
Nojus	97.7	98.1	97.1	97.9	97.1±0.78
Oskaras	99.0	100	100.0	99.4	100
Patrikas	99.0	99.8	99.8	99.8	99.8±0.20
Qju	60.1	70.8	86.0	88.1	85.6±1.63
Ričardas	90.1	93.5	87.9	96.2	91.7±1.28
Sandra	97.7	99.8	99.4	99.8	99.2±0.41
Teodoras	95.6	99.8	97.1	97.9	97.1±0.78
Ulijona	92.3	94.8	96.0	90.4	96.0±0.91
Vacys	97.7	97.3	96.7	100	98.3±0.60
Wašington	93.3	96.9	98.5	97.9	98.5±0.56
Xsas	97.5	95.0	94.6	98.9	95.4±0.97
Ygrekas	93.1	95.6	96.0	96.2	95.4±0.97
Zacharijus	94.4	93.1	96.0	29.8	96.0±0.91
Average RA % with 95% confidence intervals	**95.05±2.94**	**96.72±2.17**	**97.22±1.29**	**95.23±4.77**	**97.38±1.17**

The highest RA, 97.38±1.17%, was achieved with the male profile using mixed transcriptions.

The solution of using mixed transcriptions was formed by analyzing the correlation between the male profile and the results of word-based transcriptions, as well as the correlation between the male profile and the research results of the UPS transcriptions. The areas in Table 4.6 that are highlighted in grey indicate that this type of transcription was used in mixed transcription research with the male profile.

4.2. Acoustic modeling research

4.2.1. Lithuanian digit name recognition using a HTK-based Lithuanian recognizer

4.2.1.1. Word-based HMM recognizer research

HMM acoustic models for Lithuanian digit names were prepared using the SKAIC30 speech corpus. The database was randomly divided into two sets: a training set consisting of 24 speakers, and a test set consisting of the remaining six speakers (the distribution of speakers is presented in Annex 3). Primary investigation was performed with 1-FOLD (see Annex 3) to see how important the HMM parameters were – i.e., to identify the number of states and the number of Gaussians per state required in order to achieve a high RA.

The accuracy of digit name recognition using the REC_LTw recognizer by varying the number of states is presented in Table 4.6 and Figure 4.2.

Table 4.6. RA of Lithuanian digit names by varying number of states

	Additional number of states added to the number of states in the command, %									
	+1	**+2**	**+3**	**+4**	**+5**	**+6**	**+7**	**+8**	**+10**	**+16**
NULIS	55.0	69.2	85.0	100	100	100	100	90.8	96.7	87.5
VIENAS	100	83.3	100	92.5	96.7	99.2	95.8	98.3	99.2	97.5
DU	0	5.8	23.3	65.0	92.5	97.5	98.3	98.3	99.2	98.3
TRYS	22.5	44.2	89.2	90.8	95.0	95.0	95.0	95.0	94.2	95.0
KETURI	91.7	98.3	97.5	97.5	100	97.5	99.2	98.3	96.7	95.8
PENKI	67.5	73.3	69.2	98.3	99.2	97.5	99.2	96.7	97.5	95.8
SESI	80.8	84.2	89.2	95.8	97.5	99.2	99.2	99.2	95.8	95.8
SEPTYNI	100	100	100	100	100	100	100	100	100	99.2
ASTUONI	97.5	96.7	100	98.3	100	94.2	99.2	91.7	95.0	96.7
DEVYNI	91.7	90.0	95.0	93.3	86.7	96.7	88.3	95.8	92.5	95.0
Average RA % with 95% confidence intervals	**70.67** ± 21.67	**74.50** ± 18.23	**84.84** ± 14.64	**93.15** ± 6.44	**96.76** ± 2.70	**97.68** ± 1.24	**97.42** ± 2.25	**96.41** ± 1.93	**96.68** ± 1.48	**95.66** ± 1.98

The average accuracy of Lithuanian digit name recognition using a HTK-based recognizer was 70.67±21.67% when the number of states used in the HMMs of digit names was equal to the number of letters in the digit name plus 1 additional state (the initial state of HMM modelling). The average accuracy increased to 97.68±1.24%

when six additional states were used. By adding more than 6 states to the model, the recognition accuracy decreased progressively. The lowest RA results were achieved for the digit "du," possibly because of the fact that this digit name contains the lowest number of states.

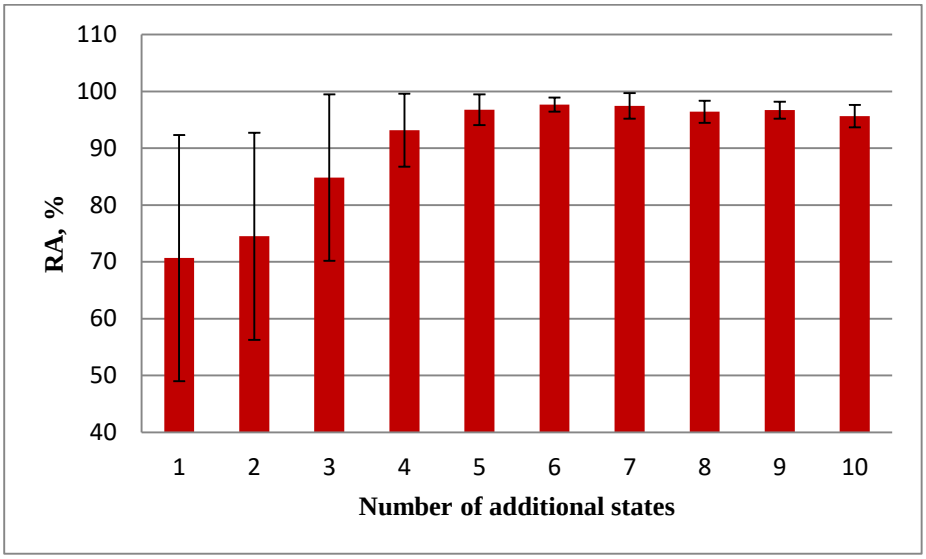


Figure 4.2. Average RA of Lithuanian digit names by number of states

By incorporating Gaussian mixtures into the Lithuanian digit name command model, recognition accuracy was significantly improved. The results of the REC_LTw recognizer by varying number of Gaussians are presented in Table 4.7.

Table 4.7. The accuracy of Lithuanian digit name recognition by varying number of Gaussian mixtures

	2 additional states, %				**4 additional states, %**				**6 additional states, %**		
	2 mix	**3 mix**	**6 mix**	**10 mix**	**2 mix**	**3 mix**	**6 mix**	**10 mix**	**2 mix**	**3 mix**	**6 mix**
NULIS	86.7	100	100	99.2	100	100	99.2	99.2	100	100	100
VIENAS	92.5	100	100	100	95	99.2	97.5	99.2	100	97.5	100
DU	11.7	75	98.3	94.2	97.5	98.3	98.3	98.3	98.3	100	98.3
TRYS	95	95	95	95	94.2	93.3	93.3	94.2	95	95	95
KETURI	97.5	100	100	100	95	95	98.3	96.7	99.2	100	96.7
PENKI	92.5	100	100	100	94.2	99.2	89.2	99.2	100	99.2	100
SESI	98.3	100	100	100	98.3	100	100	100	99.2	100	100
SEPTYNI	100	100	100	100	100	100	100	100	100	100	100
ASTUONI	90.8	100	100	100	88.3	100	100	100	100	100	100
DEVYNI	90.8	99.2	100	99.2	90.8	98.3	90	93.3	99.2	99.2	99.2
Average RA % with 95% confidence level	**85.58 ±14.42**	**96.92 ±3.08**	**99.33 ±0.67**	**98.76 ±1.24**	**95.33 ±2.35**	**98.33 ±1.45**	**96.58 ±2.59**	**98.01 ±1.53**	**99.09 ±0.91**	**99.09 ±0.91**	**98.92 ±1.08**

The highest average RA of digit names – 99.33±0.67% – was achieved with two additional states and six Gaussians. By adding 10 or more Gaussian mixtures to the models a high recognition accuracy was obtained, but this high RA could be considered an artificial accuracy because in these cases mixtures are too high. This implies a one announcer condition – each speaker establishes their own distribution. In this case, the identification process is superficial, because the recognition program cannot analyze small details. Also, doubling the number of Gaussians in turn entails doubling the demand on memory, and thus doubling the computational expense. An axial graph modeling the reliance of the research on the number of Gaussian mixtures is presented in Figure 4.3.

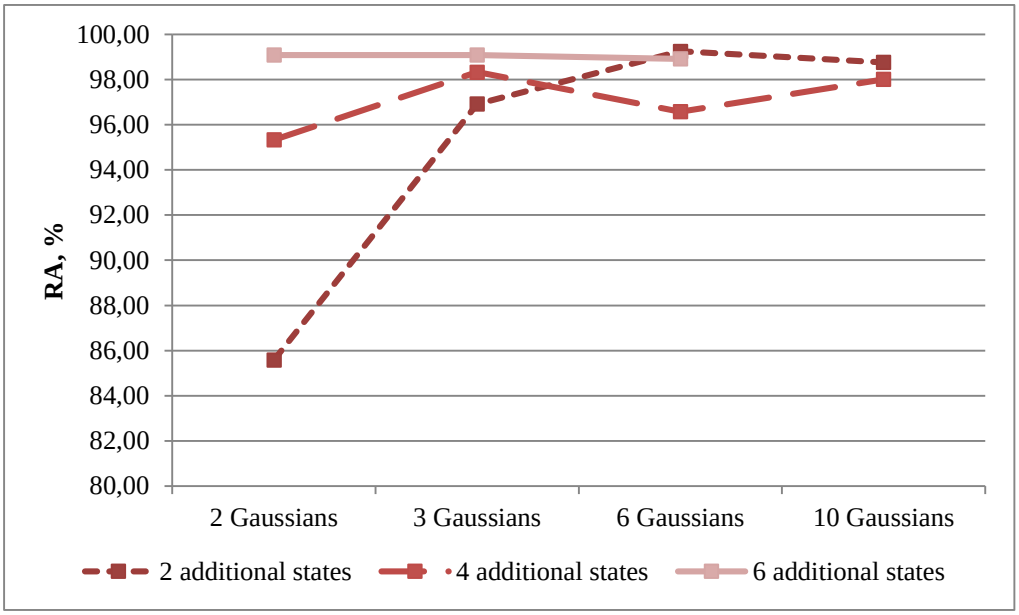


Figure 4.3. Average RA of Lithuanian digit name recognition by varying number of Gaussian mixtures

For further research, 5-times cross-validation was carried out using two additional states and six Gaussians, due to the highest RA results having been achieved with these acoustic modeling parameters. In each fold, speakers were rotated in order to test all speakers (see Annex 3).

The separate results of each digit in each fold and the individual average accuracy of each fold is presented in Table 4.8.

Table 4.8. The results of 5-times cross-validation of the RA results of Lithuanian digit names, using two additional states and six Gaussians

	1-Fold	2-Fold	3-Fold	4-Fold	5-Fold
NULIS	100	100	99.2	100	100
VIENAS	100	100	100	100	100
DU	98.3	94.2	96.7	99.2	100
TRYS	95.0	95.0	95.0	95.0	95.0
KETURI	100	100	100	100	100
PENKI	100	100	100	100	100
SESI	100	100	100	100	98.3
SEPTYNI	100	100	100	100	100
ASTUONI	100	100	100	100	100
DEVYNI	100	100	100	99.2	99.2
Average RA. % with 95% confidence level	**99.33 ±0.67**	**98.92 ±1.08**	**99.09 ±0.91**	**99.34 ±0.66**	**99.25 ±0.75**

Table 4.9 presents the average recognition accuracy of all 5 folds and each command separately. The mean recognition accuracy of all 5 folds was achieved by summing up all folds and dividing by 5. The standard deviation is 0.18.

Table 4.9. The average results of the 5-times cross-validation of the RA of Lithuanian digit names

	Average 5-times cross-validation RA, %
NULIS	99.84
VIENAS	100.0
DU	97.68
TRYS	95.0
KETURI	100.0
PENKI	100.0
SESI	99.66
SEPTYNI	100.0
ASTUONI	100.0
DEVYNI	99.68
Average RA. % with 95% confidence level	**99.19±0.81**

Most digit recognition accuracies were above 99%, but a few were lower – most notably the digits “du” and “trys” at 97.7% and 95%, respectively. Therefore, hybrid recognition technology needs to be further researched.

The result of the 5-times cross-validation of the RA results can also be compared with the one found in the thesis of Laurinciukaite (158 p. 78.), where 50 commands (10 speakers, 20 pronouncements each) from a phonetically annotated speech corpus achieved a recognition accuracy of 97.77%, with the usage of a word-based HMM and fixed values for both HMM states and number of Gaussian mixtures.

4.2.1.2. Phoneme-based HMM recognizer research

Following the technique outlined in Figure 3.8, 24 different sets of phoneme tests were established and carried out using the SKAIC30 speech corpus. As in previous tests, the entries of 24 speakers were used for the learning process, and the remaining six for testing. Some of these phonemes sets are displayed in Table 4.10, the rest are placed in Annex 4.

Table 4.10. Lithuanian digit name phoneme sets

Digit1	Digit2	Digit3	Digit5	Digit9	Digit16
Number of phonemes					
19	28 (SAMPA)	29	31	31(2)	35
Phonemes					
v	vm	vm	vm	vm	vm
ie	ie	ie	ie	-	-
n	n	n	n	n	n
a	a	a	a	a	a
s	s	s	s	s	s
d	d	d	d	d	d
u	u	u	u	u	u
t	tm	tm	tm	tm	tm
r	rm	rm	rm	rm	rm
y	y	y	y	y	y
k	km	km	km	km	km
e	e	e	e	e	e
i	i	i	i	i	i
p	pm	pm	pm	pm	pm
sh	shm	shm	shm	shm	shm
uo	uo	uo	uo	uo	uo
l	lm	lm	lm	lm	lm
sp	sil	sil	sil	sil	sil
sil	t	t	t	t	t
	ii	ii	ii	ii	ii
	nk	nk	nk	nk	nk
	sh	sh	sh	sh	sh
	nm	nm	nm	nm	nm
	dm	dm	dm	dm	dm
	sm	sm	sm	sm	sm
	sp	sp	ik	ik	ik
	ik	ik	uk	uk	ud
	uk	uk	ud	ud	ud
		ud	ish	ish	ish
			esh	esh	esh
			sp	ek	ek
				sp	en
					et
					ir
					ri
					sp

The first set, containing 19 phonemes, was the primary set. The Digit2 set, containing 28 phonemes, was selected using SAMPA phonemes. Further phoneme selection and set expansion was performed as the research progressed.

In the Digit3 set, a new phoneme – *ud* – was incorporated for command DU, which increased the command recognition of DU and the overall RA of the set from 87.02±12.98% to 94.08±5.86%. These results are displayed in Table 4.11, and detailed phoneme distribution is presented in Annex 4.

In the Digit5 set, two phonemes – *ish* and *esh* – were added to increase the command recognition accuracy of SESI. This proved to be useful, as the RA of the set increased to 94.50±5.50%.

In further research (in the Digit9 phoneme set), the phoneme *ie* was removed from the VIENAS command in order to split it into two phonemes – *i* and *ek*. This helped to increase recognition accuracy to 95.84±3.82%.

The expanded Digit16 phoneme set, containing 35 phonemes, had the highest RA of all 24 phoneme sets that were tested for recognition. During the process, additional phonemes (*en*, *et*, *ir*, and *ri*) were added to the set, which increased RA to 98.84±1.16%. Tests were carried out on other phoneme sets (Annex A), but these did not results in a significant increase in RA. The sets of phonemes tested show that the most useful are those sets which, in addition to phonemes, use softness and accent marks.

Table 4.11. Average RA results of phoneme sets

Phoneme set	**Digit1**	**Digit2**	**Digit3**	**Digit5**	**Digit9**	**Digit16**
Number of phonemes in set	19	28 (SAMPA)	29	31	31(2)	35
RA with 95% confidence intervals (1 Fold), %	**62.07 ±24.72**	**87.02 ±12.98**	**94.08 ±5.86**	**94.50 ±5.50**	**95.84 ±3.82**	**98.84 ±1.16**

From the chart presented in Figure 4.4, we are able to see the reliance of the received recognition accuracy on the number of phonemes used in sets containing from 19 to 35 phonemes. Recognition accuracy slowly increases by increasing the number of phonemes, until the maximum selected number of phonemes is reached.

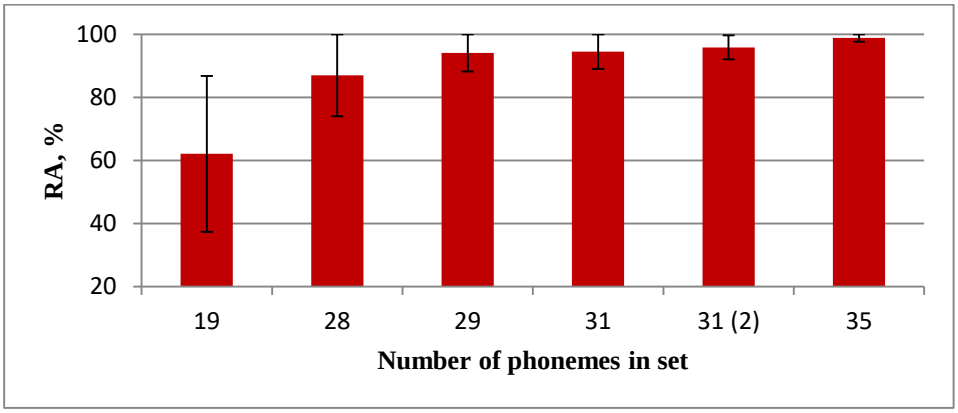


Figure 4.4. Average RA of digit names with different phoneme sets

In order to test the recordings of all speakers, a cross-validation principle (i.e., ensuring that the system of distribution of speakers in training and testing is the same as in word-based recognition research) test was carried out for the Digit1, Digit2, Digit9, and Digit16 phoneme sets. This test was performed in order to see the impact on RA of:

the primary phoneme set;

the SAMPA phoneme set;

the phoneme set selected during the process (expanded).

The abbreviated results are presented in Table 4.12, and more detailed is provided in Annex 4.

Table 4.12. Average RA results with the 5-times cross-validation of Lithuanian digit name phoneme sets

Phoneme set	Digit1	Digit2	Digit9	Digit16
Number of phonemes in set	19	28 (SAMPA)	31(2)	35
Recognition accuracy (5-times cross-validation), %	**63.05±3.59**	**84.12±2.44**	**91.65±2.94**	**97.1±1.11**

After conducting 5-times cross-validation, the highest RA was obtained using the Digit16 phoneme set – 97.1±1.11%. This result can be compared with the results of Laurinciukaite's doctoral thesis (158 p. 78), where 50 commands (10 speakers, 20 pronouncements each) from a phonetically annotated speech corpus achieved an RA of 93.91% using phoneme-based HMMs.

More detailed results of the RA of the Digit16 phoneme set, which achieved the highest result of all 24 phoneme sets, are presented in Table 4.13.

Table 4.13. The results of the 5-times cross-validation of the RA of the Digit16 Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek en et ir ri sp)

Digit16						
	Phoneme set distribution	**RA, %**				
		1-fold	**2-fold**	**3-fold**	**4-fold**	**5-fold**
VIENAS	vm i ek n a s sp	100	99.2	100	100	99.2
DU	d ud ud sp	100	90.8	95.8	100	94.2
TRYS	tm rm y s sp	95.0	94.2	95.0	95.0	92.5
KETURI	km et t u ri ir sp	100	91.7	93.3	82.5	86.7
PENKI	pm en nk km ik sp	100	94.2	100	96.7	90.0
SHESHI	shm esh shm ish sp	96.7	97.5	100	100	100
SEPTYNI	sm e pm tm y nm ik sp	96.7	100	98.3	93.3	98.3
ASHTUONI	a sh t uo nm ik sp	100	100	100	100	100
DEVYNI	dm e vm ii nm ik sp	100	98.3	99.2	100	98.3
NULIS	n uk lm i s sp	100	99.2	98.3	96.7	98.3
AVERAGE RA with 95% confidence intervals, %		**98.84 ±1.16**	**96.51 ±2.16**	**97.99 ±1.51**	**96.42 ±3.39**	**95.75 ±2.89**

Another piece of research was carried out to test whether two transcriptions for a single command would increase RA results. Three commands (SEPTYNI, ASTUONI, and DEVYNI) were chosen for this task because, when listening to the recordings, it was clear that different speakers pronounce these commands differently based on their accent. For example, SEPTYNI was pronounced either as SEPT*Y*NI or SEPTYN*I* depending on whether the speaker emphasized the vowel Y or I. The distribution of this phenomenon was evaluated, and notice was taken of the percentages of each occurrence when testing. Dict files were notated as in the following example:

SEPTYNI 0.7 sm e pm tm y nm ik sp
SEPTYNI 0.3 sm e pm tm yk nm i sp

The detailed results of this are presented in Annex A (Tables 22A–24A), but ultimately the RA of the DEVYNI command did not increase above the 99.2% result obtained from previous tests, and ASTUONI already had an RA of 100%. The command SEPTYNI instead proved that RA could decrease by having two transcriptions for one command.

These results show that command recognition accuracy depends on the proper number of phonemic segments, and as such it can be said that the selection of proper phonemic segments positively influences voice command recognition accuracy.

During the research, the number of phonemes was increased by including new phonemes which were believed to better represent obtuse sounds, or cliffs. However, increasing the number of phonemes, regardless of the features of the language, can greatly degrade RA simply because the recognizer will begin to confuse false sounds with correct sounds. Therefore, the formation of phoneme sets is a crucial task that requires extensive phonetic and phonological knowledge.

4.2.1.3. Triphone-based HMM recognizer research

Another piece of research was carried out to test models of contextual phonemes (triphones) made from monophones using HTK. Three different phoneme sets were used:

- Digit1 (the primary), containing 19 phonemes;
- Digit2 (SAMPA), containing 28 phonemes;
- Digit16 (expanded), containing 35 phonemes.

The results of this are presented in Table 4.14.

Table 4.14. RA of triphone-based HTK models

Phoneme set	Digit1	Digit2	Digit16
Number of phonemes in set	19	28	35
Recognition accuracy,%	**66.92**	**85.83**	**98.17**

Only the recognition accuracy of the primary set increased – from 62.08% to 66.92%. The RA of the other two sets decreased by using triphones due to the fact that the speech corpus was not annotated.

The early stages of triphone construction, particularly state tying, are best done with single Gaussian models (112) – therefore, further research with Gaussian mixtures was not carried out.

4.2.2. Lithuanian name and word recognition using a HTK-based Lithuanian recognizer

4.2.2.1. Word-based HMM recognizer research

HMM acoustic models for Lithuanian names and words were prepared using the NAMES3 speech corpus. The database was randomly divided into two sets: a training set consisting of 19 speakers, and a test set consisting of the remaining 3 speakers (the distribution of speakers is presented in Annex 5). Primary investigation was performed with 1-FOLD (see Annex 5) to see how important HMM parameters were – i.e., to ascertain the number of states and the number of Gaussians per state required in order to achieve a high RA.

The accuracy of name recognition using the REC_LTw recognizer by varying the number of states is presented in Figure 4.5, and the full results are presented in Annex 5, Table 3.

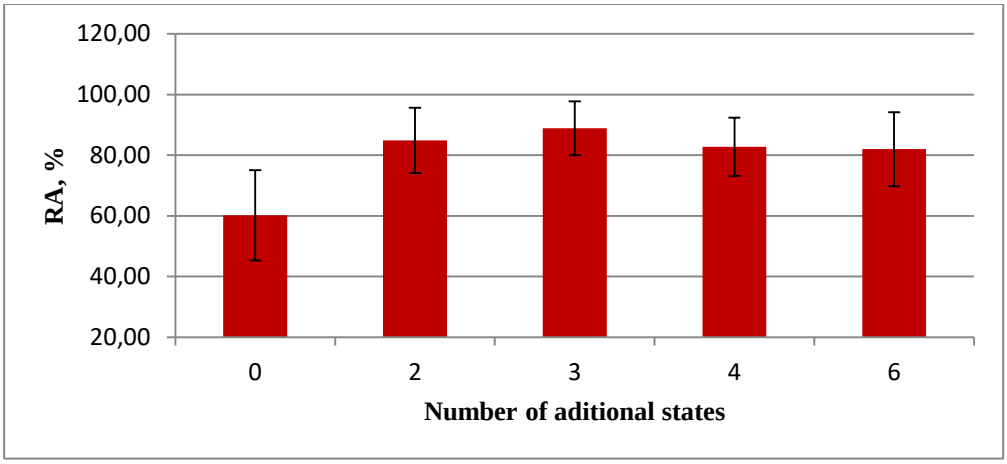


Figure 4.5. Average RA of names and words by varying number of states

The average RA of names and words using the HTK-based recognizer was 60.20±14.85% when the number of states used in the HMMs of digit names was equal to the number of letters in the name. The average accuracy increased to 88.85±8.85% when three additional states were used. By adding more than three states to the model, the recognition accuracy decreased progressively.

By incorporating Gaussian mixtures into the name command model, recognition accuracy was significantly improved. The results of the REC_LTw recognizer by varying number of Gaussians are presented in Figure 4.6, and the full list of results is provided in Annex 5, Table 5. The highest average RA of names was achieved with three additional states and two Gaussians – 99.17±0.83%.

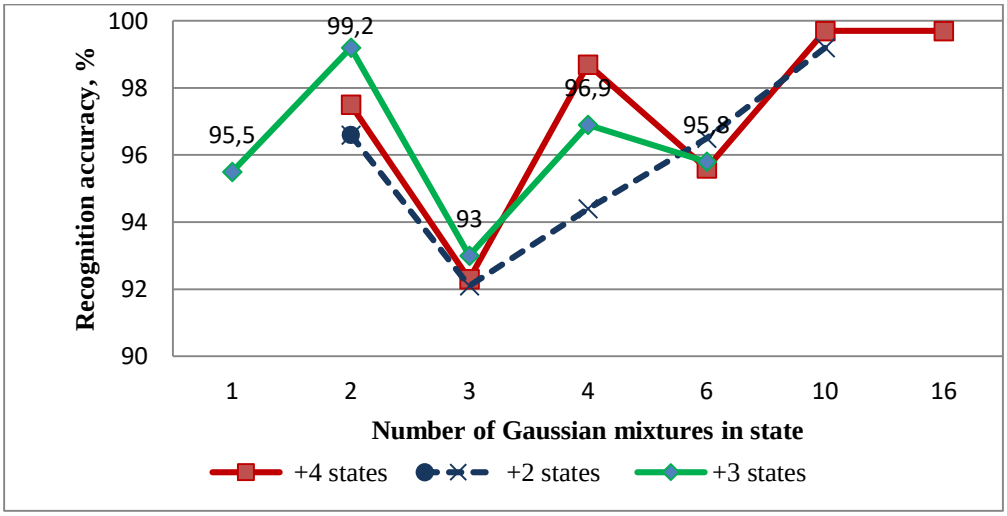


Figure 4.6. Average RA of names and words by varying number of Gaussian mixtures in states

For further research, 7-times cross-validation was carried out using three additional states and two Gaussian mixtures in state, due to the highest RA results having been achieved with these acoustic modeling parameters. The separate results of each name in each fold and the individual average accuracy of each fold is presented in Table 4.15. The average RA of all 7 folds is 96.7±2.45%

Table 4.15. RA results with 7-times cross-validation of names and words

Name	1-fold	2-fold	3-fold	4-fold	5-fold	6-fold	7-fold
Austėja	100	100.0	100	100	100	100	100
Boleslovas	100	100	100	100	100	100	100
Cecilija	100	100	100	100	100	100	100
Donatas	100	98.3	100	100	100	100	68.3
Eimantas	100	100	96.7	100	100	100	91.7
Fausta	100	96.7	100	100	100	100	98.3
Gražvydas	100	100	100	100	100	100	88.3
Hansas	100	100	95.0	96.7	98.3	78.3	90.0
Izaokas	100	100	100	100	100	100	83.3
Jonas	100	78.3	98.3	100	100	83.3	95.0
Karolis	100	100	100	100	100	100	88.3
Laima	100	100	100	100	100	100	91.7
Martynas	100	100	100	100	100	100	100
Nojus	100	71.7	98.3	60.0	96.7	88.3	100
Oskaras	100	100	100	100	100	100	90.0
Patrikas	100	93.3	100	98.3	100	100	93.3
Kju	85.0	95.0	91.7	95.0	93.3	93.3	33.3
Ričardas	98.3	100	100	100	100	100	98.3
Sandra	100	100	100	100	96.7	98.3	96.7
Teodoras	100	95.0	100	100	98.3	100	98.3
Ulijona	100	100	100	100	100	100	98.3
Vacys	98.3	100	96.7	98.3	96.7	100	98.3
Wašington	100	100	100	100	100	100	100
Xsas	93.3	96.7	95.0	71.7	66.7	76.7	33.3
Ygrekas	100	98.3	100	100	100	100	95.0
Zacharijus	100	100	100	100	100	100	100
Average RA. %	**99.03 ±0.97**	**97.05 ±2.63**	**98.91 ±0.83**	**96.92 ±3.08**	**97.95 ±2.05**	**96.85 ±2.69**	**89.60 ±7.27**

4.3. Chapter summary and results

1. The RA results of the SKAIC30 Lithuanian digit name speech corpus obtained using the REC_MSS recognizer was 99.12±0.88%, and 92.05±5.48% using the REC_SP Spanish language recognizer. Both results were obtained using transcriptions of the isolated command selection technique and mixed transcriptions.
2. Based on a selection method for names and words appropriate for the recognition of Latin letters, a speech corpus of 26 names and words was formed, which consisted of 20 utterances of each name or word by 21 speakers.
3. The RA of the name speech corpus using the REC_SP recognizer was 97.38±1.17%, obtained using a selection technique of isolated command transcriptions and mixed transcriptions.
4. Using the REC_LTw recognizer, the RA of the Lithuanian digit name speech corpus was 99.19±0.81%, and the RA of the names and words corpus was 96.7±2.45%. These results were obtained by using a technique for the recognition of isolated word commands that involved choosing the number of HMM states and Gaussian mixtures in a word-based HMM.
5. The RA of the Lithuanian digit name speech corpus with the REC_LTp phoneme-based Lithuanian language recognizer was 97.1±1.11%. This was obtained using a technique for the recognition of isolated commands by introducing new monophones into a phoneme-based HMM.

5. RESEARCH ON A HYBRID SPEECH RECOGNITION SYSTEM

The hybrid approach would make sense only if the performance of each individual approach was uncorrelated (i.e., both recognizers had a high enough recognition accuracy but their errors were largely different), or at least their performance could help in making a final decision.

Table 5.1 presents the recognition results of four different types of recognizers used for the recognition of Lithuanian digits from 0 to 9 and 26 names and words equivalent to the Latin alphabet.

Table 5.1. RAs of different recognizers using speech corpora of digits and names

Recognizer	Recognizer's name	Recognition accuracy, %	
		Digits	Names
Adapted Spanish recognizer	REC_SP	92.05	97.38
Word-based Lithuanian recognizer (isolated words)	REC_LTw	99.19	96.7
Phoneme-based Lithuanian recognizer	REC_LTp	97.1	-
Speech server (Spanish)	REC_MSS	99.12	-

Using the Weka data mining package, classification research was carried out using four different combinations of recognizers:

1. REC_LTw/REC_SP (with both the digits and the 26 names and words speech corpus)
2. REC_LTw/REC_LTp (using only the digits speech corpus)
3. REC_LTw/REC_LTp/REC_SP (using only the digits speech corpus)
4. REC_LTw/REC_MSS (using only the digits speech corpus)

Two more pieces of research on classification were carried out using the LIEPA speech corpus, with isolated words and phrases using the Kaldi toolkit. Additional research was also undertaken with a noisy corpus (NAMES3) and two different recognizers (Kaldi and TensorFlow).

A decision unit (in this case the Weka package) should realize the hybrid decision-making rule. This was taught and tested using the cross-validation method (unknown speaker).

A hybrid decision-making rule was also taught and tested using the regular 10-times cross-validation method, without regard to the interface between training objects and speakers. This was taught using 90% of the objects, the accuracy was measured using the remaining 10% of the objects, 10 tests were performed, and after changing the set of test objects the results were averaged. There were examples of the voices of the same speakers in both the training and testing samples. The results of this classification are presented in the following sections.

5.1. The connection of two recognizers: Spanish and HTK word-based

The realization of a hybrid recognizer is still an open and somewhat unresolved question. This is especially true when combining the results obtained from two recognizers that are as different as the adapted foreign language REC_SP engine and REC_LTw.

The most important part of the hybrid solution is the decision-making block. For training, we used recordings from the collected corpus when the outputs of adapted Spanish and proprietary Lithuanian recognizers differed. Using this data, the task was to create two classes (TF and FT). Each data object was formed from the decisions of two different recognizers for each utterance. This data set is characterized by significant disproportion of the objects in different classes (Tables 5.2 and 5.6).

5.1.1. Digit-name recognition: a hybrid approach

Table 5.2 summarizes the data set used to construct a decision rule for the classification of digit names by REC_LTw (two additional states and six Gaussians) and the outputs of the SP recognizer. Class TF contained 335 objects, while class FT contained only 43. A simple ("blind") decision rule – whereby if the outputs of two recognizers differ, the output of the "better" recognizer is used, in this case REC_LTw – should lead to 335/378*100 = 88.62% accuracy. Hybrid technology is therefore useful only if it surpasses this level of accuracy.

Table 5.2. Subsets of data used for decision rule training for the classifications of the REC_LTw/REC_SP recognizers

Subset	Description	Number of phrases
T=T	Both recognizers produce the same hypotheses and both hypotheses are correct	5482
F=F	Both recognizers produce the same hypotheses and both hypotheses are incorrect	-
T-	The **REC_LTw** recognizer produces a correct decision, while the **REC_SP** recognizer does not produce any decision	134
F-	The **REC_LTw** recognizer produces an incorrect decision, while the **REC_SP** recognizer does not produce any decision	3
-T	The **REC_SP** recognizer produces a correct decision, while the **REC_LTw** recognizer does not produce any decision	-
-F	The **REC_SP** recognizer produces an incorrect decision, while the **REC_LTw** recognizer does not produce any decision	-
--	Both recognizers do not produce any decision	-
TF	Both recognizers produce different hypotheses, and **REC_LTw** produces a correct decision	335
FT	Both recognizers produce different hypotheses, and **REC_SP** produces a correct decision	43
FF	Both recognizers produce different hypotheses, and both produce an incorrect decision	3
Total number of phrases		6000

The *W*-value (from the Wilcoxin signed rank test) was 7. The critical value of *W* for $N = 10$ at $p \leq 0.05$ was 8. Therefore, the result was significant at $p \leq 0.05$. There was sufficient evidence to suggest that there was a difference between the additives.

The decision-making rule was taught and tested using the 5-times cross-validation method. This was taught using the data of 24 speakers, while the data of the remaining six speakers was used for checking the accuracy of the learned rule (later, the results of the five tests were averaged). The results of this experiment are presented in Table 5.3.

A hybrid decision-making rule was also taught and tested using the regular 10-times cross-validation method, without regard to the interface between training objects and speakers. The results of this experiment results are also presented in Table 5.3.

Table 5.3. Classification accuracy results of the REC_LTw/REC_SP recognizers

Name of classifier	10-times cross-validation, %	5 times cross-validation, %
RIPPER	94.71	95.42
C4.5	96.83	95.31
Multinominal Logistic Regression	97.09	96.04
Multilayer Perceptron	96.83	96.88
ZeroR	88.62	88.22
AdaBoost	97.35	96.95
K-Nearest Neighbor (kNN)	96.56	94.77
RandomForest	98.15	98.26
Support Vector Machines	95.50	93.45
NaiveBayes	92.06	85.83

From the analysis of the results provided in Table 5.3, it is evident that the ZeroR classifier is inappropriate for the task set due to it having achieved the lowest results. The best classification results in both cases were obtained using the RF classifier (100 trees). After changing the random seed for XVal/%Split from 1 to 40, which specifies the random seed used when randomizing the data before it is divided up for evaluation purposes, the average classification accuracy with 10-times cross-validation was 98.15 with standard deviation of 0.24.

A hybrid decision-making rule learned with the RF classifier works 99.79±0.07% of the time when 10-times cross-validation is performed, and 99.78% of the time when the speaker is not known and 5-times cross-validation is performed. Compared to the REC_LTw recognizer alone, the error percentage of the 10-times cross-validation results decreased by 72.84%.

Based on the results presented by Jovic and Bogunovic (172), it is appropriate to look for the number of trees for the RF classifier that is most efficient in the sense of classification accuracy.

For this, a 10-times cross-validation experiment was performed using 90% of the objects for training, and using the remaining 10% of the objects for testing and changing the number of trees of the RF classifier from 1 to 100. The results of this classification are presented in Figure 5.1.

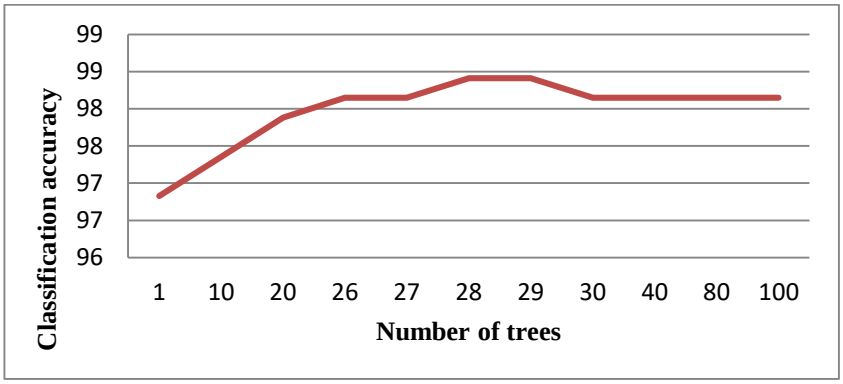


Figure 5.1. The reliance of classification accuracy on the number of trees in the RF classifier

By increasing the number of trees, classification accuracy rises slightly until it reaches a maximum of 98.41%, when the number of trees is equal to 28 and 29. Further increase in the number of trees does not result in an increase in accuracy, as from 30 to 100 tress accuracy remains stable at 98.15%.

Using the Weka data mining package, an additional test was performed to analyze the impact of features such as "REC_LTw_prob," "REC_SP_prob," "REC_SP_supp," "REC_LTw_delta," "letters" (32 letters in the vocabulary of 10 digit names), and "gender." For this, a 10-times cross-validation experiment was carried out by eliminating certain features and using an RF classifier (28 trees). The results of this experiment are presented in Table 5.4.

Table 5.4. The reliance of classification accuracy results on features with the RF classifier

Feature list	Classification accuracy, %
Full list (REC_LTw_prob, REC_SP _prob, Class, REC_SP _supp, REC_LTw_delta, letters, gender)	**98,41**
Full list (no letters)	93.65
Full list (no gender)	98.15
Full list (no_REC_LTw_delta)	98.68
Full list (no REC_SP_supp)	98.41
Full list (no REC_SP _supp, no_REC_LTw_delta)	96.83
Full list (no gender, no letters)	94.97
Full list (no letters, no REC_LTw_delta)	92.59
Full list (no gender, no_ REC_SP _supp)	98.15
Full list (no letters, no_ REC_SP _supp)	94.18
Full list (no gender, no_REC_LTw_delta)	98.41

Only REC_LTw_prob, REC_SP _prob, Class	92.06
Only REC_LTw_prob, REC_SP _prob, Class, letters	98.15
Only REC_LTw_prob, REC_SP _prob, Class, gender	93.39
Only REC_LTw_prob, REC_SP _prob, Class, REC_LTw_ delta	94.44
Only REC_LTw_prob, REC_SP _prob, Class, REC_SP _supp	93.39

The highest classification accuracy results were achieved with the *Full list (no_REC_LTW_delta)* feature set, however the unmodified *Full list* achieved a lower classification accuracy. It can be seen from the results that the "letters" feature has the most impact on classification accuracy and the "SP_supp" feature has no impact on classification accuracy.

5.1.2. Lithuanian name and word recognition: a hybrid approach

Two recognizers – REC_LTw (three additional states and two Gaussians) and REC_SP – were used for the isolated word command recognition of 26 names and words. The results obtained were gathered and systemized for classification, and the subsets and number of phrases in each is presented in Table 5.5. The "blind" decision rule was also applied which, as subset FT had more objects, in this case means that the "better" recognizer would be the REC_SP recognizer, and should lead to an accuracy of 314/611*100 = 51.39%. If classification accuracy were to surpass this result, then hybrid technology would be useful.

Table 5.5. Subsets of data used for decision rule training for the classifications of the REC_LTw/REC_SP recognizers of the NAMES3 speech corpus

Subset	Description	Number of phrases
T=T	Both recognizers produce the same hypotheses and both hypotheses are correct	10,253
F=F	Both recognizers produce the same hypotheses and both hypotheses are incorrect	19
T-	Recognizer **REC_LTw** produces correct decision while recognizer **REC_SP** does not produce any decision	1
F-	The **REC_LTw** recognizer produces an incorrect decision, while the **REC_SP** recognizer does not produce any decision	1
-T	The **REC_SP** recognizer produces a correct decision, while the **REC_LTw** recognizer does not produce any decision	-
-F	The **REC_SP** recognizer produces an incorrect decision, while the **REC_LTw** recognizer does not produce any decision	-
--	Both recognizers do not produce any decision	-
TF	Both recognizers produce different hypotheses, and **REC_LTw** produces a correct decision	297
FT	Both recognizers produce different hypotheses, and **REC_SP** produces a correct decision	314
FF	Both recognizers produce different hypotheses, and both produce an incorrect decision	35
Total number of phrases		10,920

The *W*-value was 170. The critical value of *W* for $N = 26$ at $p \leq 0.05$ was 98. Therefore, the result was not significant at $p \leq 0.05$.

As in previous classification research, the decision-making rule was taught and tested in two ways: the 10-times cross-validation method and the 7-times cross-validation method. Classification was taught using the data of 18 speakers, while the data of the remaining three speakers was used for checking the accuracy of the learned rule (later on, the results of seven tests were averaged). These test results are presented in Table 5.6.

Table 5.6. Classification accuracy results of the REC_LTw/REC_SP recognizers

Name of classifier	**10-times cross-validation mean, %**	**7-times cross-validation mean, %**
RIPPER	95.91	89.25
C4.5	99.02	90.69
Multinominal Logistic Regression	93.45	85.39
Multilayer Perceptron	97.71	90.32
ZeroR	51.39	42.21
AdaBoost	93.78	90.37
K-Nearest Neighbor (kNN)	96.07	86.67
Random Forest	99.02	95.26
Support Vector Machines	96.07	87.07
NaiveBayes	85.11	83.81

The random forest classifier managed to achieve the highest accuracy of the ten classifiers listed. After changing the random seed for XVal/%Split from 1 to 40, the average classification accuracy with 10-times cross-validation was 98.93% with a standard deviation of 0.27.

A hybrid decision-making rule learned by an RF classifier works with 99.44±0.09% accuracy when a speaker is known, and 99.23% when a speaker is not known. Compared to the REC_LTw recognizer alone, the error percentage of the results decreased by 70.61%.

The number of trees used at testing was 100. Therefore, research was carried out to evaluate the dependency on the number of trees in the RF classifier. The results of this are presented in Figure 5.2.

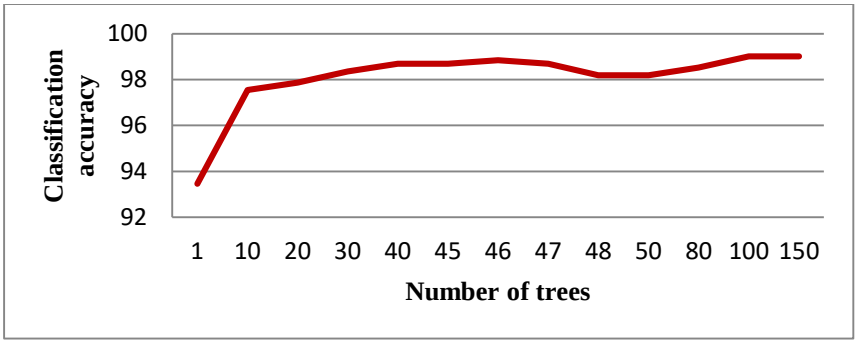


Figure 5.2. The reliance of classification accuracy on the number of trees in the RF classifier

Depending on the varying number of trees in the RF classifier, accuracy fluctuated from 98.85 with 46 trees, to 98.19 with 48 trees, and then recovering to a stable 99.02 with 100–150 trees.

Further research was carried out to evaluate the reliance of classification accuracy on the features of the names vocabulary, using the RF classifier with 46 trees. The number of features used for the research was 62, depending mostly on the number of letters distributed in commands. The results obtained are presented in Table 5.7.

Table 5.7. The reliance of classification accuracy results on features with the RF classifier

Feature list	Classification accuracy, %
Full list	**98.85**
Full list(no letters)	91.82
Full list(no gender)	98.85
Full list(no_REC_LTw_delta)	98.36
Full list (no REC_SP_supp)	98.36
Full list(no REC_SP_supp,no_REC_LTw_delta)	96.89
Full list (no gender, no letters)	91.65
Full list(no letters,no_REC_LTw_delta)	86.91
Full list(no gender,no_ REC_SP_supp)	96.89
Full list(no letters,no_ REC_SP_supp)	91.49
Full list(no gender,no_REC_LTw_delta)	97.71
REC_LTw_prob, REC_SP_prob, Class	76.10
REC_LTw_prob, REC_SP_prob, Class, letters	95.42
REC_LTw_prob, REC_SP_prob, Class, gender	78.39
REC_LTw_prob, REC_SP_prob, Class, REC_LTw_delta	91.33
REC_LTw_prob, REC_SP_prob, Class, REC_SP_supp	85.11

After analysis of the test results, and taking account into the possibilities of realization, it is evident that the inclusion of some of the features in the classification is not essential, as they exert little or no influence on the classification result. For example, the “gender” feature had no influence on the accuracy of the classification result, though the determination of gender in speech recognition is a difficult task.

5.2. The connection of two recognizers: word-based and phoneme-based HTK

The following research was carried out by combining the REC_LTw (two additional states and six Gaussians) and REC_LTp (extended 35 phoneme set) recognizers using the digit vocabulary. Although the recognizers are both HTK-based, the produced recognition hypotheses differ due to their different methods. The distribution of subsets is presented in Table 5.8. In the case of this classification, the "blind" decision rule was again used. The TF subset had more objects and the "better" recognizer was the REC_LTw recognizer, which should lead to an accuracy of 138/148*100 = 93.24%.

Table 5.8. Subsets of data used for decision rule training with the classification of the results of the REC_LTw and REC_LTp recognizers

Subset	Description	Number of phrases
T=T	Both recognizers produce the same hypotheses and both hypotheses are correct	5,816
F=F	Both recognizers produce the same hypotheses and both hypotheses are incorrect	23
T-	The **REC_LTw** recognizer produces a correct decision, while the **REC_LTp** recognizer does not produce any decision	-
F-	The **REC_LTw** recognizer produces an incorrect decision, while the **REC_LTp** recognizer does not produce any decision	-
-T	The **REC_LTp** recognizer produces a correct decision, while the **REC_LTw** recognizer does not produce any decision	-
-F	The **REC_LTp** recognizer produces an incorrect decision, while the **REC_LTw** recognizer does not produce any decision	-
--	Both recognizers do not produce any decision	-
TF	Both recognizers produce different hypotheses, and **REC_LTw** produces a correct decision	138
FT	Both recognizers produce different hypotheses, and **REC_LTp** produces a correct decision	10
FF	Both recognizers produce different hypotheses, and both produce an incorrect decision	13
Total number of phrases		6,000

The *W*-value was 1, and the critical value of *W* for $N = 10$ at $p \leq 0.05$ was 8. Therefore, the result was significant at $p \leq 0.05$. There was sufficient evidence to suggest that there was a difference between the additives.

The REC_LTw/REC_LTp hybrid approach has the most phrases in the subset of F=F, which will influence the hybrid decision-making rule percentage. The classification accuracy results of the REC_LTw and REC_LTp recognizers are presented in Table 5.9.

Table 5.9. Classification accuracy of REC_LTw and REC_LTp with different classifiers

Name of classifier	10-times cross-validation mean, %	5 times cross-validation mean, %
RIPPER	96.62	96.71
C4.5	97.97	97.31
Multinominal Logistic Regression	99.32	99.13
Multilayer Perceptron	98.65	90.24
ZeroR	93.24	89.83
AdaBoost	99.32	96.71
K-Nearest Neighbor (kNN)	100	99.13
RandomForest	99.32	99.13
Support Vector Machines	100	99.13
NaiveBayes	100	97.31

After changing the random seed for XVal/%Split from 1 to 40, the average classification accuracy using the 10-times cross-validation method did not change with either classifier.

Although classification accuracy was 100%, when 10-times cross-validation was used a hybrid decision-making rule learned by the KNN and SVM classifiers worked with 99.40% accuracy due to the F = F subset and the number of words unrecognized by both recognizers. With the results of an unknown speaker, a hybrid decision-making rule was learned 99.38% of the time. Compared to the results of the REC_LTw recognizer, the error percentage decreased by only 23.46%.

Research with the kNN classifier was performed by varying the number of neighbors from 1 to 100. These results are presented in Figure 5.3.

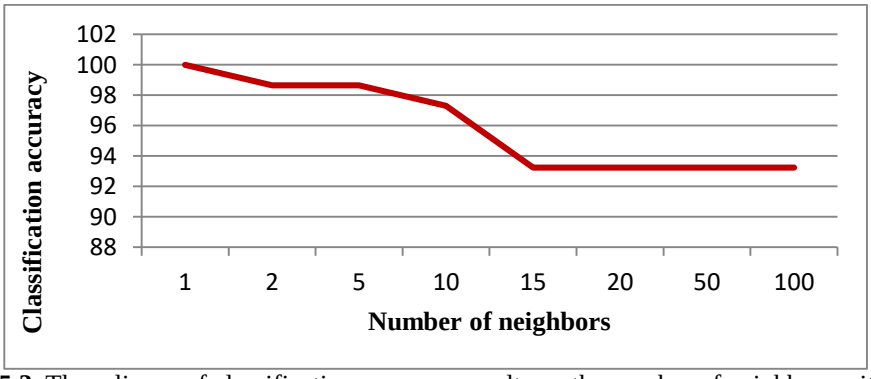


Figure 5.3. The reliance of classification accuracy results on the number of neighbors with the kNN classifier

Figure 5.3 shows that by increasing the number of neighbors with the kNN classifier, classification accuracy decreases. A classification accuracy of 100% was achieved using one neighbor. This accuracy stabilizes at 93.24% from 15 neighbors to 100.

Another piece of research was performed with the SVM classifier, by varying kernels. The results of this are presented in Table 5.10.

Table 5.10. The reliance of classification accuracy on the kernels used in the SVM classifier

Kernel	**Accuracy of Correctly Classified Instances, %**
PolyKernel	100
NormalizedPolyKernel	100
Puk	97.30
RBFKernel	100

Because two classifiers presented the same classification accuracy, research into reliance on features was carried out using two classifiers: kNN (one neighbor) and SVM (PolyKernel). The results of both classifiers are presented in Table 5.11.

Table 5.11. The reliance of classification accuracy results on features, with the kNN and SVM classifiers

Feature list	**Classification accuracy, %**	
	kNN	**SVM**
Full list (REC_LTw_prob, REC_LTp_prob, REC_LTP_supp, REC_LTw_delta, letters, gender)	100	100
Full list (no letters)	93.24	93.24
Full list (no gender)	100	100
Full list (no_REC_LTw_delta)	100	100
Full list (no REC_LTp_supp)	100	100
Full list (no REC_LTp_supp, no_REC_LTw_delta)	100	100
Full list (no gender, no letters)	92.57	93.24
Full list (no letters, no REC_LTw_delta)	93.24	93.24
Full list (no gender, no_REC_LTp_supp)	100	100
Full list (no letters, no_REC_LTp_supp)	95.27	93.24
Full list (no gender, no_REC_LTw_delta)	100	100
Only REC_LTw_prob, REC_LTp_prob, class	95.27	93.24
Only REC_LTw_prob, REC_LTp_prob, letters	100	100
Only REC_LTw_prob, REC_LTp_prob, gender	95.95	93.24
Only REC_LTw_prob, REC_LTp_prob, REC_LTW_ delta	95.27	93.24
Only REC_LTw_prob, REC_LTP_prob, REC_LTP_supp	93.24	93.24

In this research into reliance on features, slightly better classification results were achieved using the kNN classifier. It was evident that using only the "REC_LTw_prob," "REC_LTp_prob," and "letters" features provides the same classification accuracy as using the full list of features.

5.3. The connection of three recognizers: Spanish, HTK word-based, and HTK phoneme-based

In order to create a hybrid speech recognizer, it was first necessary to find the best combinations of speech recognizers that produced the highest RA results.

Therefore, three recognizers – REC_LTw (two additional states and six Gaussians), REC_LTp (extended 35 phoneme set), and REC_SP – were combined using the digit vocabulary. The distribution of these subsets is presented in Table 5.12.

Table 5.12. Subsets of data used for decision rule training for the classification of the results of the REC_LTw/REC_LTp/REC_SP recognizers

Subset	Description	Number of phrases
TTT	All recognizers produce the same hypotheses and all hypotheses are correct	5,362
FFF	All recognizers produce the same or different hypotheses and all hypotheses are incorrect	1
TTF	The **REC_LTw** and **REC_SP** recognizers produce the same hypotheses and the hypotheses are correct, the **REC_LTp** hypothesis is incorrect	118
TFT	The **REC_LTw** and **REC_LTp** recognizers produce the same hypotheses and the hypotheses are correct, the **REC_SP** hypothesis is incorrect	451
FTT	The **SP** and **REC_LTp** recognizers produce the same hypotheses and the hypotheses are correct, the **REC_LTw** hypothesis is incorrect	8
TFF	The **SP** and **REC_LTp** recognizers produce the same hypotheses and the hypotheses are incorrect, the **REC_LTw** hypothesis is correct	20
FTF	The **REC_LTw** and **REC_LTp** recognizers produce the same hypotheses and the hypotheses are incorrect, the **REC_SP** hypothesis is correct	35
FFT	The **REC_LTw** and **REC_SP** recognizers produce the same hypotheses and the hypotheses are incorrect, the **REC_LTp** hypothesis is correct	5
Total number of phrases		6,000

When evaluating the Kruskal–Wallis H test, the H statistic was 9.42 (2, $N = 30$). The p-value was 0.009, and the result was significant at $p \leq 0.05$. The recognition accuracy was significantly different among the three recognizers.

Subset FFF, where all recognizers produced the same or different hypotheses and the hypotheses were incorrect, had only one phrase. This means that there was only one case where all of the recognizers produced an incorrect answer – in all other cases, one or two recognizers produced the correct answer. The results of this classification are presented in Table 5.13.

The highest classification results were obtained by the RF classifier (100 trees). After changing the random seed for XVal/%Split from 1 to 40, the average classification accuracy with 10-times cross-validation was 99.67% (standard deviation 0.28).

Table 5.13. Classification accuracy results of the REC_LTw / REC_LTp / REC_SP recognizers

Name of classifier	10-times cross-validation mean, %	5 times cross-validation mean, %
RIPPER	96.23	94.17
C4.5	95.91	95.102
Multinominal Logistic Regression	96.54	93.27
Multilayer Perceptron	96.70	95.67
ZeroR	92.46	92.69
AdaBoost	92.46	93.04
K-Nearest Neighbor (kNN)	96.86	95.95
RandomForest	97.17	96.52
Support Vector Machines	96.07	95.08
NaiveBayes	92.77	93.45

The hybrid decision-making rule was evaluated by adding together the number of phrases of the subsets TTF+TFT+FTT+TFF+FTF+FFT and multiplying it by the classification accuracy, and adding the number of phrases in the TTT subset and dividing by the total number of phrases. The classifier hybrid decision-making rule worked at an accuracy of 99.67±0.09% with 10-times cross-validation, and at 99.61% with 5-times cross-validation. Compared to the REC_LTw recognizer alone, the error percentage of these results decreased by 51.85%.

Using the RF classifier, reliance on number of trees was also evaluated in the classification of these three recognizers. This classification accuracy is presented in Figure 5.4.

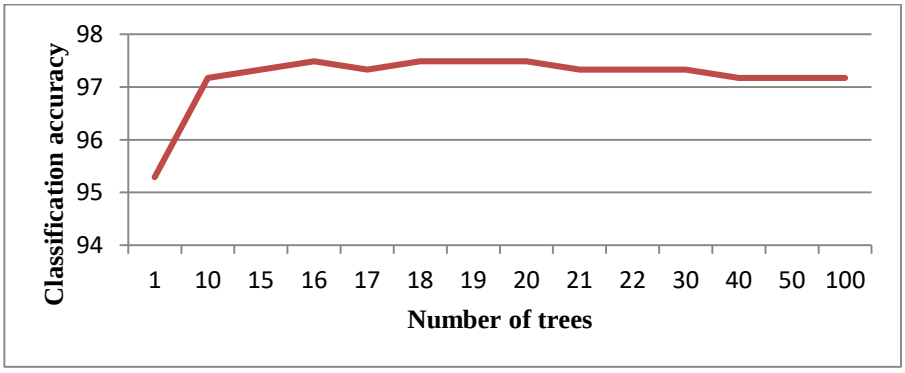


Figure 5.4. The reliance of classification accuracy on the number of trees in the RF classifier

5.4. The connection of two recognizers: HTK word-based and Speech Server

Another hybrid solution was evaluated using the REC_LTw (two additional states and six Gaussians) and REC_MSS recognizers with the digit vocabulary. The subsets and numbers of phrases in each subset are presented in Table 5.14.

Table 5.14. Subsets of data used for decision rule training for the classifications of REC_LTw/REC_MSS

Subset	Description	Number of phrases
T=T	Both recognizers produce the same hypotheses and both hypotheses are correct	5,902
F=F	Both recognizers produce the same hypotheses and both hypotheses are incorrect	-
T-	The **REC_LTw** recognizer produces a correct decision, while the **REC_MSS** recognizer does not produce any decision	3
F-	The **REC_LTw** recognizer produces an incorrect decision, while the **REC_MSS** recognizer does not produce any decision	-
-T	The **REC_MSS** recognizer produces a correct decision, while the **REC_LTw** recognizer does not produce any decision	-
-F	The **REC_MSS** recognizer produces an incorrect decision, while the **REC_LTw** recognizer does not produce any decision	-
--	Both recognizers do not produce any decision	-
TF	Both recognizers produce different hypotheses, and **REC_LTw** produces a correct decision	51
FT	Both recognizers produce different hypotheses, and **REC_MSS** produces a correct decision	44
FF	Both recognizers produce different hypotheses, and both produce an incorrect decision	-
Total number of phrases		6,000

The *W*-value was 13.5. The critical value of *W* for $N = 7$ at $p \leq 0.05$ was 2. Therefore, the result was *not* significant at $p \leq 0.05$.

Using the Weka package and ten data mining algorithms, the classification accuracy results of these two recognizers are presented in Table 5.15.

Table 5.15. Classification accuracy results of the REC_LTw/REC_MSS recognizers

Name of classifier	**10-times cross-validation mean, %**	**5-times cross-validation mean, %**
RIPPER	95.79	89.58
C4.5	90.53	78.16
Multinominal Logistic Regression	97.89	81.55
Multilayer Perceptron	98.95	81.25
ZeroR	53.68	45.46
AdaBoost	95.79	92.08
K-Nearest Neighbor (kNN)	96.84	89.88
RandomForest	100	93.33
Support Vector Machines	94.74	80.29
NaiveBayes	93.68	80.16

The highest classification results in both cases were obtained by the RF classifier (100 trees). After changing the random seed for XVal/%Split from 1 to 40, the average classification accuracy with 10-times cross-validation was 99.66% (standard deviation 0.60). A hybrid decision-making rule learned by the RF classifier works with 99.99±0.01% accuracy when 10-times cross-validation is applied, and 99.89% when 5-times cross-validation is applied. Compared to the REC_LTw recognizer alone, the error percentage of these results decreased by 86.42%.

The reliance of classification accuracy on the number of trees with the RF classifier is presented in Figure 5.5.

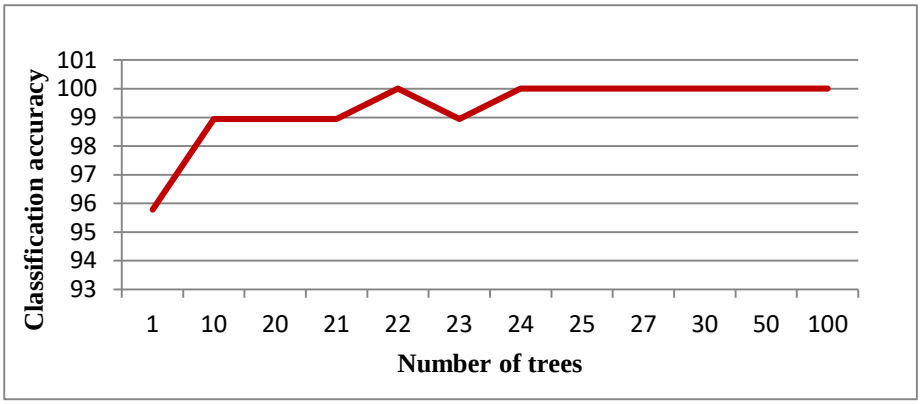


Figure 5.5. The reliance of classification accuracy on the number of trees in the RF classifier

The highest classification accuracy was obtained with 22 trees. The accuracy then decreased to 98.95% with 23 trees before climbing again using 24 trees, remaining stable until 100 trees.

Therefore, an analysis of the reliance of the classification accuracy on the features of the RF classifier was carried out using 22 trees. These results are presented in Table 5.16.

Of the features used for classification, a "Full list" was most effective. The highest impact on results was exerted by the "letters" feature. By eliminating this feature, the classification accuracy decreased by 23.16%. The lowest impact was exerted by the features of "gender" and "REC_LTw_delta."

Table 5.16. The results of the reliance of classification accuracy on features with the RF classifier

Feature list	Classification accuracy, %
Full list (REC_LTw_prob, REC_MSS_prob, Class, REC_MSS _supp, REC_LTW_delta, letters, gender)	**100**
Full list (no letters)	76.84
Full list (no gender)	98.95
Full list (no_REC_LTw_delta)	98.95
Full list (no REC_MSS_supp)	94.74
Full list (no REC_MSS_supp, no_REC_LTw_delta)	95.79
Full list (no gender, no letters)	81.05
Full list (no letters, no REC_LTw_delta)	77.89
Full list (no gender, no_ REC_MSS_supp)	96.84
Full list (no letters, no_ REC_MSS_supp)	78.95
Full list (no gender, no_REC_LTw_delta)	95.79
Only REC_LTw_prob, REC_MSS_prob, Class	76.84
Only REC_LTw_prob, REC_MSS_prob, Class, letters	95.79
Only REC_LTw_prob, REC_MSS_prob, Class, gender	71.58
Only REC_LTw_prob, REC_MSS_prob, Class, REC_LTw_ delta	75.79
Only REC_LTw_prob, REC_MSS_prob, Class, REC_MSS_supp	77.89

5.5. The recognition of the LIEPA speech corpus using a hybrid approach

5.5.1. Isolated word recognition

A 5-times cross-validation test was carried out using two additional states and six Gaussians due to the highest RA results having been achieved with these acoustic modeling parameters during the REC_LTw research in subsection 4.2.1.1. In each fold, speakers were rotated in order to test all 50 of them (see Annex 6). The results of this analysis are presented in Table 5.17.

Table 5.17. The results of the 5-times cross-validation of the RA of Lithuanian digit names from the LIEPA speech corpus, using two additional states and six Gaussians

Command	1-Fold	2-Fold	3-Fold	4-Fold	5-Fold
NULIS	100	60	100	100	100
VIENAS	100	40	100	100	100
DU	80	100	90	100	90
TRYS	70	60	100	100	90
KETURI	90	80	100	90	100
PENKI	90	50	80	100	100
SESI	100	90	100	100	100
SEPTYNI	100	90	90	100	90
ASTUONI	100	90	100	100	100
DEVYNI	100	90	100	100	100
Average RA with 95% confidence level, %	**93±6.57**	**75±12.82**	**96±4**	**99±1**	**97±2.99**

The average recognition accuracy of all 5 folds was 92±3.31% (standard deviation 5.33).

The second recognizer, REC_SP, was used with the same fragment of the LIEPA speech corpus. A male profile and mixed transcriptions were used due to the highest RA results having been achieved under these conditions in the previous testing of REC_SP in section 4.1.2.1. The results of this are presented in Table 5.18.

Table 5.18. RA of digit names with REC_SP using the LIEPA speech corpus

Command	**REC_SP RA, %**
NULIS	78
VIENAS	96
DU	90
TRYS	82
KETURI	46
PENKI	62
SESI	84
SEPTYNI	88
ASTUONI	90
DEVYNI	92
Average RA, %	**80.8±11.09**

The average RA of digit names with REC_SP using the LIEPA speech corpus was 80.8±9.61% (standard deviation 5.33).

The obtained results were gathered and systemized for classification, and the subsets and numbers of phrases in each subset are presented in Table 5.19.

Table 5.19. The subsets of data used for decision rule training for the classification of the results of the REC_LTW/REC_SP recognizers with the LIEPA speech corpus

Subset	Description	Number of phrases
T=T	Both recognizers produce the same hypotheses and both hypotheses are correct	375
F=F	Both recognizers produce the same hypotheses and both hypotheses are incorrect	3
T-	The REC_LTw recognizer produces a correct decision, while the REC_SP recognizer does not produce any decision	64
F-	The REC_LTw recognizer produces an incorrect decision, while the REC_SP recognizer does not produce any decision	6
-T	The REC_SP recognizer produces a correct decision, while the REC_LTw recognizer does not produce any decision	-
-F	The REC_SP recognizer produces an incorrect decision, while the REC_LTw recognizer does not produce any decision	-
--	Both recognizers do not produce any decision	-
TF	Both recognizers produce different hypotheses, and REC_LTw produces a correct decision	21

FT	Both recognizers produce different hypotheses, and REC_SP produces a correct decision	29
FF	Both recognizers produce different hypotheses, and both produce an incorrect decision	2
Total number of phrases		500

The W-value was 1. The critical value of W for $N = 10$ at $p \leq 0.05$ was 8. Therefore, the result was significant at $p \leq 0.05$. There was sufficient evidence to suggest that there was a difference between the additives.

The decision-making rule was taught and tested with the 10-times cross-validation method. The results of this experiment are presented in Table 5.20.

Table 5.20. The results of the classification accuracy of the REC_LTw/REC_SP recognizers

Name of classifier	**10-times cross-validation mean, %**
RIPPER	82
C4.5	82
Multinominal Logistic Regression	70
Multilayer Perceptron	88
ZeroR	60
AdaBoost	82
K-Nearest Neighbor (kNN)	88
RandomForest	86
Support Vector Machines	82
NaiveBayes	82

Of the 10 classifiers listed, the Multilayer Perceptron and kNN classifiers managed to achieve the highest accuracy. After changing the random seed for XVal/%Split from 1 to 40, the average classification accuracy with 10-times cross-validation was 87.35% (standard deviation 2.19) with the Multilayer Perceptron classifier, and 87.8% (standard deviation 1.96) with kNN.

With 10-times cross-validation testing, a hybrid decision-making rule was learned 96.74±0.68% of the time with the Multilayer Perceptron classifier.

Compared to the results of the REC_LTw recognizer alone, the error percentage of these results decreased by 59.25% with the Multilayer Perceptron classifier.

These results show that the acoustic models and connection technology examined are suitable for digit recognition with different speech corpora.

5.5.2. Phrase recognition

Phrase recognition research was carried out using the LIEPA speech corpus. The Z060 corpus was selected for this research, due to its large number of speakers and the majority of its commands being phrases (see Table 3.2). The part that was used for recognition contained 143 speakers and 26 commands (eight isolated words and

18 phrases containing up to four words). This research was executed to show that not only isolated words could be recognized with this method.

Using the Kaldi toolkit (57), recognition research was carried out using acoustic models involving both monophones and triphones. The same MFCC-based features as in the HTK package and the other default parameters of monophone and triphone recognition methods were used for phrase recognition by the Kaldi package. Approximately 20% of the corpus was used as the test set, and the remainder was used for training the system. The 5-times cross-validation method was carried out with the whole Z060 corpus. The recognition results are presented in Table 5.21.

Table 5.21. RA results of monophone and triphone acoustic models of phrases in the LIEPA speech corpus

Models	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	**AVERAGE RA, %**
Monophone	89.44	79.75	89.92	87.15	83.90	**86.04±8.25%**
Triphone	88.02	88.71	90.46	91.09	91.45	**89.95±4.25%**

The average phrase recognition accuracy with monophone acoustic models was 86.04% while using default parameters, and 89.95% with triphone models by adding Gaussian mixtures. Compared to word recognition accuracy, these results were lower, as monophone and triphone acoustic modelling returned results of 91.99% and 93.92%, respectively. Kaldi gives an average estimate of the logarithmic probability of the whole phrase, and HTK gives the logarithmic probability of each word individually. For these reasons, it was decided to evaluate the recognition accuracy of the whole phrase.

There were no cases where either recognizer did not produce a decision, but there were many where both recognizers produced the same incorrect decision. Most of these incorrect decisions were missing the short word "į" – e.g., "grįžti į skyriaus turinį," "grįžti į temos turinį." Many mistakes were also made in failing to recognize another short word: "ir" – e.g., "junesko ir lietuva."

The results obtained were gathered and systemized for classification, and the complementarity of the results of both recognizers is presented in Table 5.22.

Table 5.22. The complementarity of monophone- and triphone-based recognizers

Subset	Description	Number of phrases
T=T	Both recognizers produced the same correct decision	2,900
F=F	Both recognizers produced the same incorrect decision	144
T-	The recognizer with triphone AM produced a correct decision, while the recognizer with monophone AM did not produce a decision	-
F-	The recognizer with monophone AM produced a correct decision, while the recognizer with triphone AM did not produce a decision	-

-T	The recognizer with monophone AM produced a correct decision, while the recognizer with triphone AM did not produce a decision	-
-F	The recognizer with triphone AM produced a correct decision, while the recognizer with monophone AM did not produce a decision	-
--	Both recognizers did not produce a decision	-
TF	Only the recognizer with triphone AM produced a correct decision	159
FT	Only the recognizer with monophone AM produced a correct decision	302
FF	Both recognizers produced different incorrect decisions	64

The decision-making rule was taught and tested with the 10-times cross-validation method. The features used for classification were: obtained logarithmic probability from recognizers output, letters (the proportion of the number of certain letters to the number of all letters in the decision of the recognizers), and gender. The results of the classification experiment are presented in Table 5.23.

Table 5.23. The classification accuracy of the results of recognizers

Name of classifier	**10-times cross-validation mean, %**
RIPPER	91.32
C4.5	92.19
Multilayer Perceptron	91.54
ZeroR	65.51
AdaBoost	85.25
RandomForest	94.14
NaiveBayes	83.30

The highest classification results were obtained by the RF classifier. After changing the random seed for XVal/%Split from 1 to 40, the average classification accuracy with 10-times cross-validation was 93.07% (standard deviation 0.5). A hybrid decision-making rule learned by an RF classifier works with 93.44±0.15% accuracy when the 10-times cross-validation test is applied.

Compared to those of the triphone acoustic model recognizer alone, the error percentage of these results decreased by 34.73%.

5.6. The connection of two different recognition engines and the use of a noisy speech corpus

The previous experiments were executed using no additional signal processing. No artificial noise was added to the signal, but in order to examine if the model could be applied to real life, signal processing was necessary. To generate such data, 5 dB of white noise was added to the audio signal.

Using the Kaldi toolkit (57), recognition research was carried out using triphone acoustic models. The same MFCC-based features as in the HTK package and the other default parameters of the triphone recognition method were used.

Deep Speech 2 was selected as the different recognition engine.

With both of the previously mentioned recognizers, the NAMES3 speech corpus (21 speakers, 26 commands, 20 utterances) was trained and tested with 7 folds, using data from 18 speakers for training and 3 for testing the system. The triphone-based recognizer was taught and tested using a phoneme set with diphthongs.

As previously mentioned, 7 folds were taught and tested with both recognizers, the results of which are presented below in Table 5.24.

Table 5.24. The recognition accuracy results of triphone-based and Deep Speech 2 recognizers

Recognizer	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	**Average RA,%**
Triphone-based	90.96	88.32	90.57	89.29	91.73	93.27	79.29	**88.75±3.40%**
Deep Speech 2	85.96	85.04	93.26	84.49	85.38	88.85	70.32	**84.56±5.23%**

A higher average recognition accuracy of 88.75% was achieved with the Kaldi package using triphone acoustic models. Deep Speech 2 with RNN was able to achieve an RA of only 84.56%. This might be because default settings were used for this experiment. However, the goal of this experiment was to determine if the connection of these two recognizers would obtain higher RA results than the RA of one or the other recognizer alone. Results obtained from both recognizers are summarized in Table 5.25.

Table 5.25. The complementarity of the results of triphone-based and Deep Speech 2 recognizers

Subset	Description	Number of phrases
T=T	Both recognizers produced the same hypotheses and both hypotheses were correct	8,534
F=F	Both recognizers produced the same hypotheses and both hypotheses were incorrect	30
T-	The triphone-based recognizer produced a correct decision while the Deep speech 2 recognizer did not produce a decision	0
F-	The triphone-based recognizer produced an incorrect decision while the Deep speech 2 recognizer did not produce a decision	0
-T	The Deep speech 2 recognizer produced a correct decision while the triphone-based recognizer did not produce a decision	11
-F	The Deep speech 2 recognizer produced an incorrect decision while the triphone-based recognizer did not produce a decision	15
--	Both recognizers did not produce a decision	0
TF	Both recognizers produced different hypotheses, and the triphone-based recognizer produced a correct decision	1,118
FT	Both recognizers produced different hypotheses, and the Deep speech 2 recognizer produced a correct decision	639
FF	Both recognizers produced different hypotheses, and both produced an incorrect decision	496
Total number of phrases		10,843

In this experiment, there were cases when the triphone-based recognizer did not produce a decision but the Deep Speech 2 recognizer produced a correct decision. In this case, the hybrid decision rule submits the correct answer thanks to the Deep Speech 2 recognizer. There were also cases when the triphone-based recognizer did not produce a decision and Deep Speech 2 produced an incorrect decision, so these samples are accepted as incorrect in the hybrid decision rule.

The decision-making rule was taught and tested with the 10-times cross-validation method. The features used for classification were obtained using logarithmic probability from triphone-based recognizer output, letters (the proportion of the number of certain letters to the number of all letters in the decision of the recognizer), and gender.

The random forest classifier was used for the connection of both recognizers on the basis of the earlier experiment. After changing the random seed for XVal/%Split from 1 to 40, the average classification accuracy with 10-times cross-validation was 92.62% (standard deviation 0.32). A hybrid decision-making rule learned by the RF classifier worked with 93.81±0.1% accuracy when the 10-times cross-validation test was applied.

The W-value was 37.5. The critical value of W for $N = 26$ at $p \leq 0.05$ was 89. Therefore, the result was significant at $p \leq 0.05$. There was sufficient evidence to suggest that there was a difference between the additives. The root mean squared error was 0.2407, which demonstrates that the model can predict the data relatively accurately. Compared to the results of the triphone-based recognizer alone, the error percentage decreased by 44.44%.

5.7. The recognition of the INFOBALSAS speech corpus using a hybrid approach

The REC_LTtri Lithuanian speech recognizer is based on a CD-HMM model. Its basic version uses triphones as a basic speech element to model acoustic events that occur during speech recording. Gaussian mixtures are used to model the probabilities of particular acoustic events, and acoustic properties are described using MFCC features. The Viterbi search algorithm was used as the basis for the decoding procedure to find the most likely sequences of acoustic events (88).

The output of the Lithuanian REC_LTtri recognizer presented separate logarithmic probabilities for each word in a phrase. An example output is presented below:

"fdanbru/LIGOS/ANKI_SPO/d1010.rec"
1100000 9600000 ankilozinis -68.571236
9600000 18800000 spondilitas -65.389702

The mean logarithmic probability of the whole phrase was calculated according to the length of the frame, e.g., “ankilozinis spondilitas” – 66.918. This logarithmic probability was used as one of the features for the connection of recognizers

The recognition results of 731 voice commands from the MEDIC medical speech corpus were used in the construction of a hybrid recognizer. All of the results

obtained from both recognizers were grouped into several subsets. These subsets are summarized in Table 5.26 (123).

Table 5.26. The complementarity of the results of the REC_LTtri and REC_SP recognizers

Subset	Description	Number of phrases
1	Both recognizers produced the same correct decision	135,898
2	Both recognizers produced the same incorrect decision	178
3	The REC_LTtri recognizer produced a correct decision, while the REC_SP recognizer did not produce a decision	3,398
4	The REC_LTtri recognizer produced an incorrect decision, while the REC_SP recognizer did not produce a decision	48
5	The REC_SP recognizer produced a correct decision, while the REC_LTtri recognizer did not produce a decision	7
6	The REC_SP recognizer produced an incorrect decision, while the REC_LTtri recognizer did not produce a decision	1
7	Both recognizers did not produce a decision	1
8	Only REC_LTtri produced a correct decision	33,650
9	Only REC_SP produced a correct decision	1,357
10	Both recognizers produced different incorrect decisions	902

The Weka package was selected for research into the connection of the two recognizers. A hybrid decision-making rule was taught and tested with the 12-times cross-validation method. This was taught using the data of 11 speakers, while the data of the remaining speaker was used for checking the accuracy of the learned rule (later on, the results of these 12 tests were averaged). The experiment showed that the set of decision-making rules learned by RIPPER works with 97.85±2.30% accuracy. Because the decision rule is called into action only when the REC_SP and REC_LTtri solutions differ, the average operating accuracy of the hybrid recognizer is 98.92%

A hybrid decision-making rule was also taught and tested with the regular 10-times cross-validation method, without regard to interface between training objects and speakers. This rule was taught using 90% of the objects, with its accuracy measured using the remaining 10% of the objects; 10 tests were performed, and after changing the set of test objects the results were averaged. There were examples of the voices of the same speakers in the training and testing samples. The 10-times cross-validation method showed that the hybrid decision rule learned by RIPPER works with 98.73±0.24% accuracy, and the "blind" decision rule accuracy was 96.12%. Thus, the ATP_HB hybrid recognizer correctly recognizes all 1 subset records (135,898), all 3 subtype records (3,398), and recognizes 8 + 9 subset records (34,562 out of 35,007) with 98.73% accuracy. This means that the average operating accuracy of the hybrid recognizer is 99.10% – (135,898 + 3,398 + 34,562) / 175,440. This result is valid when ATP_HB recognizes the speech of one of the 12 known speakers.

The RIPPER classifier was selected for the implementation of the hybrid decision-making rule because it provides a very simple set of rules. An example of these rules is presented below:

SP :- lt_prob<=-73.44, lt_space>=10, lt_d>=10, sp_a<=14.3
default LT

The set of rules of the RIPPER algorithm is arranged and applied in turn: if the first rule is not suitable, the second rule is applied, and so on. The REC_SP recognizer rule lists cases in which it is worth believing the decision of the REC_SP recognizer rather than that of the REC_LTtri recognizer. If none of the above rules apply, then the last "default LT" rule recommends believing the decision of the REC_LTtri recognizer. In almost all REC_SP rules, the conjunctive lt_delta_prob <= threshold is present. This means that the decision of the REC_SP recognizer is offered to be used if the REC_LTtri recognizer is not completely sure of its proposed priority solution.

The RIPPER classifier and other such hybrid recognizers decrease recognition error by 24% compared with a HTK-based Lithuanian recognizer alone. This is another result that demonstrates that the hybrid recognizer is suitable for the recognition of phrases.

5.8. Hybrid recognition technology

A voice-controlled, web-service-based prototype of a hybrid recognizer was developed during the INFOBALSAS project using the proposed method of recognizer connection. It can be described as a set of internet objects and functions for accessing remote-based service operations. All calculations are performed server-side, and users are presented with an HTML5-based frontend compatible with any modern browsers and devices, including Android-based phones and tablets. This client-server principle allows the full control, support, and improvement of speech recognition processes, and also reduces the calculations performed on a client's device, as speech processing is very computationally intensive. The web service itself was developed using .NET4 WCF libraries and is compatible with industry standard applications (88).

The main recognition process can be explained in three steps:

1) A user pronounces a voice prompt using their device of choice, thus a sound recording is produced and sent to the server for further processing;

2) As soon as the server receives the recorded audio file, the signal processing components are activated and the recording is then further passed to both speech recognizers, which then continue the recognition process and produce the possible semantic meanings and probabilities of a recognized answer;

3) To make the final decision rule, the RIPPER induction algorithm (173) is applied due to its simplicity: a set of rules found using the RIPPER algorithm are arranged. This means that rules are applied in a given order: if the first rule cannot be applied, the second rule should be applied, and so on. The response is generated and sent back to the client's device (application or web script) via an encrypted string and is then further shown on screen, passed for further application steps, or even pronounced using a proprietary Lithuanian TTS.

The highest classification accuracy from the 10 most popular data mining algorithms for digit names and Lithuanian names was achieved by the random forest (RF) classifier. The realization of RF is more complicated compared to the RIPPER algorithm, but it can be done using manuals and packages (174).

5.9. Chapter summary and results

1. By connecting two or three recognizers using the Lithuanian digit name speech corpus, the highest RA was achieved by the hybrid of REC_LTw/REC_SP after using the 10-times cross-validation method: 99.79±0.07%. The total error decrease with the hybrid of REC_LTw/REC_SP recognizers was 72.84%.
2. After connecting the REC_LTw and REC_MSS recognizers, the results were the highest of all digit speech corpus recognition tests with a hybrid recognizer made for a GSM signal. The achieved RA was 99.99% with the 10-times cross-validation test, and the decrease in recognition failure was 86.42%.
3. Using the names speech corpus, the highest RA was achieved with a hybrid of the REC_LTw/REC_SP recognizers: 99.44% with 10-times cross-validation, and 99.23% with 7-times cross-validation.
4. Research with part of the LIEPA the speech corpus showed that the method of connection could be performed with other speech corpora. The RA achieved by the hybrid of the REC_LTw/REC_SP recognizers was 96.74±0.68% with 10-times cross-validation.
5. Research on the recognition and connection of phrases was carried out using part of the LIEPA speech corpus. This demonstrated that the hybrid method could also be applied to phrases. Although the recognition accuracy achieved was no higher than 95%, compared to a triphone acoustic model recognizer alone the error percentage of these results decreased by 34.73%.
6. Research was performed with two different speech recognition engines (Microsoft and Baidu) using the names speech corpus with a signal: noise ratio of 5 dB. The connection of results improved recognition accuracy by 44.44% compared to the use of a triphone-based recognizer alone.
7. In all cases, when connecting recognizers it was determined that hybrid recognizers were more accurate than separate recognizers.
8. Of the 10 classifiers used to determine the hybrid recognizer connection rule, the best results were achieved in four cases out of five with the random forest classifier. In the other case, the kNN and SVM classifiers achieved an equal score, which tied them for the best results.
9. By increasing the number of trees in the RF classifier, classification accuracy slightly rose until it reached its maximum. Further increase in the number of trees did not result in an increase in accuracy.
10. The impact of features on classification accuracy showed that the main features were the confidence measure of the REC_SP recognizer and the average log probability of the REC_LTw or REC_LTp recognizers.
11. The classification results achieved by REC_LTw/REC_SP (99.81±0.07%) can be compared with those found in the work of Rasymas and Rudžionis (131), published in 2015. There, an RA of 98.16% was achieved from a speech corpus

(50 commands, 12 speakers, 20 utterances each) by connecting five recognizers (Lithuanian, Russian, English, and two German.)

12. Research from the INFOBALSAS project provides an example that a hybrid approach to speech recognition could be applied for the recognition of not only isolated words, but also of phrases. In this project, a hybrid recognizer decreased recognition error by 24% compared with a HTK-based Lithuanian recognizer alone.

6. CONCLUSIONS

1. The SKAIC30 speech corpus of ten digit names (30 speakers, 10 digits, 20 pronouncements of each digit) and the NAMES3 speech corpus of Lithuanian names and words (21 speakers, 26 names or words, 20 pronouncements of each name or word) were collected, and were used for investigations into their recognition. It was found that the name and word selection technique created ensured very high recognition accuracy (RA) – 97.38±1.17% – of the NAMES3 speech corpus by the adapted REC_SP Spanish recognizer. For comparison, the RA of the NATO alphabet using the same recognizer was only 67.2% (two speakers, 26 words, 50 pronouncements of each word).
2. The REC_SP Spanish language recognizer (8.0 (Spanish-US)) was selected as the non-native recognizer, and MSS'2007 Spanish language recognizer (9.0 for MSS (Spanish-US)) (REC_MSS) was selected for telephone applications. Tests on the recognition of the SKAIC30 speech corpus by the REC_MSS recognizer showed that the isolated command transcription selection technique allowed for the attainment of a very high RA – 99.12±0.88% – for this corpus.
3. The REC_LTw Lithuanian language recognizer, with a word-based HMM, and the REC_LTp Lithuanian language recognizer, with a phoneme-based HMM, were created and investigated. The technique for the recognition of isolated word commands by choosing the number of HMM states and Gaussian mixtures in the word-based HMM allowed for the attainment of a very high RA – 99.19±0.81% – of the SKAIC30 speech corpus by the REC_LTw recognizer. By using the technique of introducing new monophones into a phoneme-based HMM, a 97.1±1.11% RA of the SKAIC30 speech corpus by the REC_LTp recognizer was achieved, even without using the phonetic segmentation of the speech corpus.
4. The results of research into connecting two or three recognizers showed that the suggested method of using machine learning for the connection of different recognizers improved the RA of the speech corpora used in all cases:
5. By connecting two or three recognizers using the SKAIC30 speech corpus, the best RA result was achieved by a hybrid of the REC_LTw/REC_SP recognizers – 99.78%. The most significant RA error decrease was also achieved by a hybrid of the REC_LTw/REC_SP recognizers – 72.84%, when the 5-times cross-validation average method was used.
6. After connecting the REC_LTw and REC_MSS recognizers, their results were the highest of all digit speech corpus recognition tests with a hybrid recognizer made for a telephone signal. The RA achieved was 99.89% with the 5-times cross-validation test, and the decrease in recognition failure was 86.42%.
7. Using the NAMES3 name speech corpus, the RA achieved by a hybrid of the REC_LTw/REC_SP recognizers was 99.44±0.09% with the 10-times cross-validation test, and 99.23% with the 7-times cross-validation test.
8. The hybrid recognizer decreased the recognition error of the MEDIC medical speech corpus by 24% compared with a HTK-based Lithuanian recognizer alone. Of this speech corpus, 52.26% of recordings were phrases containing two to five words.

9. Research on part of the LIEPA speech corpus has shown that the developed acoustic models and method of connecting recognizers work with both phrases and other speech corpora.
10. Research with different recognition engines and added noise has shown that the connection of recognizers achieves better results than one recognizer alone, and that this method works with different engines and noisy signals.
11. The proposed method of connecting recognizers was implemented in new hybrid recognition technology, created and proved during the INFOBALSAS project.
12. The results obtained can be compared with the results of different Lithuanian authors:
    - The RA of the adapted non-native recognizer is significantly higher than the 92.5% presented in the thesis of Maskeliūnas (83 p. 111). This RA was achieved using a digit speech corpus with the following parameters: 10 speakers, 10 digits, and 20 pronouncements of each digit.
    - The REC_LTw Lithuanian language recognizer with a word-based HMM can be compared with the one found in the thesis of Laurinčiukaitė (158 p. 78). There, the RA of 50 commands was 97.77% (31 speakers, 20 utterances of each command).
    - The RA results of the REC_LTp Lithuanian language recognizer with a phoneme-based HMM can be compared with the 93.91% achieved in the doctoral thesis of Laurinčiukaitė (158 p. 78). This result was attained using phoneme-based HMMs and a speech corpus with the following parameters: 31 speakers, 50 commands, 20 pronouncements of each command.
    - The results of the combination of recognizers achieved in this thesis can be compared with those found in the work of Rasymas and Rudžionis (131), published in 2015. There, an RA of 98.16% was achieved from a speech corpus (50 commands, 12 speakers, 20 utterances each) by connecting five recognizers (Lithuanian, Russian, English, and two German).

7. SANTRAUKA

Šnekamoji kalba yra kasdienio bendravimo priemonė. Sparčiai tobulėjant technologijoms ir joms užimant vis svarbesnę vietą kasdienėje žmonių veikloje, tampa labai aktualu pritaikyti technologijas taip, kad būtų įmanoma jas valdyti žmonėms pačiu priimtiniausiu būdu – balsu. Todėl pagrindinis automatinio šnekos atpažinimo sistemų kūrėjų ir tobulintojų tikslas yra sukurti technologijas, kurios galėtų girdėti, suprasti, kalbėti ir veikti pagal balsu gautą informaciją.

Pastaruoju metu šnekos atpažinimo technologijos yra plačiai taikomos informacinėse technologijose. Todėl šnekos atpažinimo priemonių ir metodų pritaikymas informacinėse technologijose yra viena labiausiai tyrinėjamų sričių. Šnekamosios kalbos atpažinimo sistemos taikomos automobilių pramonėje (laisvų rankų įranga, navigacijos bei multimedijos prietaisų valdymas balsu), mobiliuosiuose telefonuose ir daug kitų įvairių sričių. Sėkmingi tyrimai šnekos atpažinimo srityje reikalauja didelių finansinių išteklių ir gausaus duomenų rinkinio, apimančio ir sudėtingas balso komandas. Žinoma kompanija *Google* sėkmingai vykdo tokius tyrimus, paremtus paslėptųjų Markovo modelių (PMM) principu, ir stebina rezultatais – sistemos atpažįsta net neplačiai vartojamų kalbų žodžius. Šios įmonės sėkmę lemia surinkti dideli garsynai, naudojami šnekos atpažinimo sistemoms mokyti.

Lietuvių kalba nėra plačiai vartojama pasaulyje, todėl kitos šalys nesuinteresuotos skirti daug dėmesio jos tyrimams ir šnekos atpažinimo sistemų pritaikymui. Tačiau Lietuvos mokslininkai ir tyrėjai gali sėkmingai pritaikyti jau esamus šnekos atpažinimo produktus saviems tyrimams.

Darbo tikslas

Disertacijos tikslas – sukurti hibridinę lietuviškų balso komandų atpažinimo technologiją sujungiant du ar daugiau šnekos atpažintuvų. Tikimasi, jog tuo atveju, kai vienas iš sujungtų atpažintuvų suklys, kitas ar kiti priims teisingą sprendimą. Pasirinktas hibridinio atpažintuvo taikymas – iš raidžių ir skaitmenų sudarytų kodų atpažinimas per mikrofoną, taip pat skaitmenų kodo atpažinimas per telefoną.

Darbo uždaviniai

Darbo tikslui pasiekti išsikelti šie uždaviniai:

1. Surinkti skaičių pavadinimų ir vardų garsynus, tinkamus kodams, susidedantiems iš skaičių ir lotyniškų raidžių, atpažinti.
2. Kitakalbį atpažintuvą pritaikyti lietuviškoms balso komandoms atpažinti.
3. Paruošti du lietuvių šnekos atpažintuvus taikant žodžiais grįstus ir fonemomis grįstus PMM.
4. Sujungti du ir daugiau atpažintuvų taikant mašininio mokymo metodus.
5. Gautus atpažinimo tikslumo tyrimo rezultatus palyginti su kitų Lietuvoje atliktų tyrimų rezultatais.

Tyrimų metodika ir priemonės

Pasirinktas hibridinis atpažintuvo modelis, nes sujungus kelias skirtingas atpažinimo sistemas, veikiančias pagal skirtingus metodus, padidinamas balso

komandų atpažinimo tikslumas. Pasirinktas hibridinio atpažintuvo taikymas – kodų atpažinimas per mikrofoną ir per telefoną. Lietuviškas atpažintuvas buvo modeliuojamas su HTK programinių įrankių paketu pagal žodžių, fonemų ir kontekstinių fonemų PMM akustinius modelius. Pasirinkti PMM MFCC (melo dažnių kepstro koeficientai) požymiai, užtikrinantys gerus izoliuotų komandų atpažinimo rezultatus. Mikrofoniniam taikymui pasirinktas su operacinėmis sistemomis *Windows'7* ir *Windows'8* laisvai platinamas ispanų šnekos atpažintuvas *Microsoft Speech Recognizer 8.0 (Spanish-US*), o telefoniniam taikymui – balso serveryje *Microsoft Speech Server (MSS'2007)* naudojamas ispanų šnekos atpažintuvas *Microsoft Speech Recognizer 9.0 for MSS (Spanish-US*). Abiem atpažintuvams sujungti pasirinktas laisvai platinamas WEKA programų paketas.

Lietuviškų vardų atrankos metodika sukurta remiantis lietuviškų vardų atpažinimo tikslumo tyrimų su ispanų šnekos atpažintuvu *Microsoft Speech Recognizer 8.0 (Spanish-US)* rezultatais.

Darbo mokslinis naujumas

1. Sukurta vardų ir kitokių žodžių atrankos metodika, tinkanti lotyniškoms raidėms identifikuoti atpažįstant siūlomus vardus ar kitokius žodžius. Metodika užtikrina daugiau nei 30 % didesnį lotyniškų raidžių atpažinimo tikslumą, palyginti su NATO abėcėlės garsyno atpažinimo tikslumu.

2. Pasiūlyta kelių atpažintuvų sujungimo taikant mašininį mokymą metodika. Jos skiriamasis bruožas yra požymių, gautų iš atpažintuvų, sujungimas su papildomais požymiais, priklausančiais nuo atpažinto žodžio. Metodika patikrinta šiais atvejais:

- sujungiant keturių skirtingų garsynų ar jų fragmentų atpažinimo rezultatus:

a) dviejų skaičių pavadinimų garsynų (30 diktorių, 10 skaičių pavadinimų po 20 ištarimų ir 50 diktorių, 10 skaičių pavadinimų po 1 ištarimą),

b) vardų garsyno (21 diktorius, 26 vardai arba kitokie žodžiai, atitinkantys tam tikrą abėcėlės raidę, po 20 ištarimų),

c) medicinos terminų garsyno (731 frazė arba žodis, 12 diktorių, po 20 ištarimų);

d) frazių ir žodžių garsyno (146 diktoriai, 18 frazių, 8 žodžiai po 1 ištarimą);

- sujungiant su skirtingais atpažinimo varikliais (*Microsoft* ir *Baidu*) gautus vardų garsyno atpažinimo rezultatus;
- sujungiant 5 dB lygyje užtriukšminto vardų garsyno atpažinimo rezultatus;
- sujungiant telefoninio formato (8 kHz, 8 bitai) skaičių pavadinimų garsyno atpažinimo rezultatus.

Atpažinimo tyrimuose naudoti trys programų paketai: HTK, *Kaldi* ir *TensorFlow*.

Darbo rezultatų praktinė reikšmė

Disertacijos tyrimų rezultatai galėtų būti naudojami kuriant lietuvių kalbos automatinio šnekos atpažinimo sistemas, paremtas kodų atpažinimu. Kodai, sudaryti iš raidžių ir skaitmenų, galėtų būti panaudoti ligų pavadinimams (TLK-10-AM), prekių kodams, PIN kodams ir kt. atpažinti per mikrofoną. Taip pat taikytinas iš skaičių sudarytų kodų atpažinimas per telefoną ir mikrofoną.

Siūlomas atpažintuvų sujungimo metodas buvo pritaikytas hibridinio atpažinimo technologijoje, sukurtoje ir patvirtintoje projekto „INFOBALSAS“ metu.

Ginamieji teiginiai

1. Siūloma vardų ir kitokių žodžių, tinkamų lotyniškoms raidėms identifikuoti, metodika užtikrina daugiau nei 30 % didesnį vardų ir kitokių žodžių garsyno (21 diktorius, 26 vardai ir kitokie žodžiai, 20 ištarimų) atpažinimo tikslumą, palyginti su NATO abėcėlės garsyno atpažinimo tikslumu (2 diktoriai, 26 žodžiai, 50 ištarimų).

2. Siūloma kelių atpažintuvų sujungimo metodika (mašininio mokymosi metodo taikymas derinant iš atpažintuvų gautus požymius bei naudojant papildomus požymius, kurie priklauso nuo atpažinto žodžio) leido pagerinti visų tyrimams naudojamų šnekos garsynų atpažinimo tikslumą. Pagrindiniai šios metodikos aspektai:

- klasifikavimo procese naudojami visi šnekos garsyno požymiai. Tai pagrindinis skirtumas nuo kelių kitų atpažintuvų sujungimo būdų;
- tyrimuose naudojamus papildomus požymius (sp_supp, lt_delta_prob, gender, lt_a,…, lt_ž, sp_a, ..., sp_ž) sugeneruoja kalbos ekspertai, naudodamiesi atpažintuvų išvestimis arba rankiniu būdu. Šie požymiai visais tirtais atvejais leido padidinti klasifikavimo tikslumą, palyginti su klasifikavimo naudojant požymius, gautus vien iš atpažintuvų, rezultatais;
- dviejų ar trijų atpažintuvų sujungimo tyrimai, kuriuose buvo naudojami keli šnekos garsynai, parodė, kad siūlomas metodas visais nagrinėtais atvejais padidina garsynų atpažinimo tikslumą;
- siūloma kelių atpažintuvų sujungimo metodika buvo išbandyta naudojant medicininį šnekos garsyną, susidedantį iš atskirų žodžių ir frazių. RIPPER klasifikatorius ir siūlomas hibridinis atpažintuvas atpažinimo klaidų skaičių sumažina 24 %, palyginti su vien HTK pagrindu veikiančiu lietuvišku atpažintuvu.

Dalyvavimas projektuose

Dalyvauta Aukštųjų technologijų plėtros 2011–2013 metų programos projekte „Hibridinė atpažinimo technologija balso sąsajai (INFOBALSAS)“.

7.1. Lietuviškų balso komandų atpažinimo problemos

Atlikus literatūros analizę paaiškėjo, kad plačiausiai taikomas izoliuotų komandų atpažinimo metodas yra paslėptieji Markovo modeliai. Izoliuotoms komandoms atpažinti taikomi žodžiais, fonemomis arba kontekstinėmis fonemomis grįsti PMM. Kaip atpažinimo požymiai paprastai naudojami signalo energija, melų skalės kepstro koeficientai ir jų pirmosios bei antrosios eilės išvestinės. Požymių vektoriaus ilgis lygus 39. Gilieji neuroniniai tinklai turi pranašumą prieš PMM, bet norint juos naudoti Lietuvoje reikia papildomų išteklių, visų pirma, finansavimo.

Vienas svarbiausių šnekos atpažinimo sistemos elementų yra garsyno surinkimas ir anotavimas. Tam reikia daug žmogiškųjų išteklių ir laiko. Neturint garsynų, anotuotų fonemų lygmeniu, izoliuotoms komandoms atpažinti turėtų būti taikomi žodžiais grįsti PMM, nes jiems pakanka lengvai realizuojamo garsyno segmentavimo žodžių lygmeniu. Kaip paaiškėjo atlikus literatūros analizę, užsienyje garsynų atpažinimo tikslumo rezultatai lyginami su rezultatais, gautais naudojant tuos

pačius garsynus. Lietuvoje tokių galimybių nėra dėl lietuvių kalbos specifikos ir tyrimuose naudojamų skirtingų garsynų, todėl atliktų tyrimų rezultatus nutarta lyginti tik su lietuvių tyrėjų gautais rezultatais.

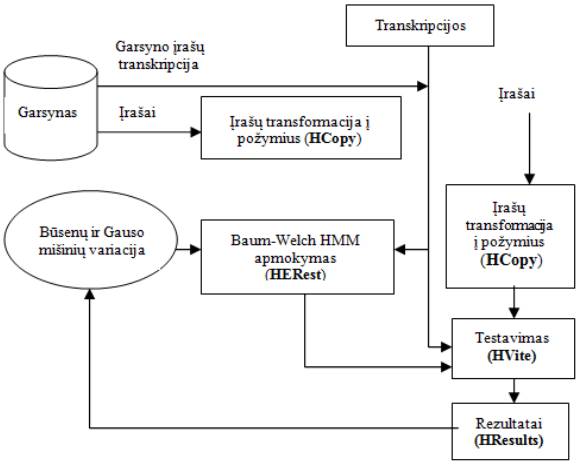
Hibridinis atpažintuvas veikia šiuo principu: vienu metu naudojami du skirtingi atpažintuvai. Tai prasminga, jei, vienam atpažintuvui klystant, kitas priima teisingą sprendimą. Kartu su lietuviškais atpažintuvais, veikiančiais pritaikius žodžiais ir fonemomis grįstus PMM, hibridinio atpažintuvo tyrimams pasirinktas ispanų šnekos atpažintuvas, o telefoniniam taikymui – *Microsoft* balso serveris, leidžiantis kurti interaktyvaus balso atsakymo per telefoną sistemas, ir šio serverio ispanų šnekos atpažintuvas.

7.2. Tyrimų metodika ir priemonės

7.2.1. Tyrimuose naudoti ištekliai ir priemonės

Šiuo metu dažniausiai taikomas matematinis balso atpažinimo modelis – paslėptosios Markovo grandinės (PMM). Jose randama tikėtiniausia ištartoji balso komanda (atskiras žodis ar žodžių seka), atitinkanti tam tikrus parinktus parametrus ir tenkinanti tam tikrus apribojimus (113). Atpažinimo modelio parametrai gaunami mokymo metu, o mokoma pagal įvairių diktorių balso įrašus (garsynus).

PMM technologija buvo taikoma lietuviškiems atpažintuvams REC_LTw ir REC_LTp sukurti. Atpažintuvų akustiniams modeliams sudaryti buvo naudojamas atvirojo kodo programinių priemonių rinkinys HTK v.3.2 (*Hidden Markov Toolkit*) (112). Žodžiais grįsto PMM akustinių modelių sudarymo ir testavimo procesą iliustruoja 7.1 pav. pateikta schema.

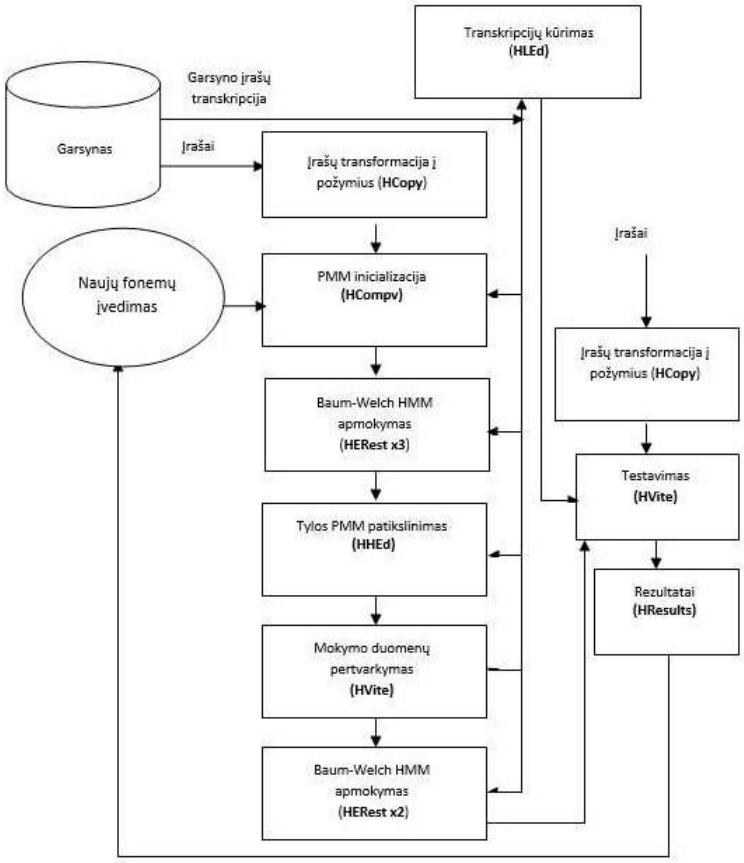


7.1. pav. Žodžiais grįsto PMM akustinių modelių sudarymo procesas

Taikant HTK programinių įrankių paketą balso komandoms modeliuoti fonemų metodu reikia specialiai paruošti duomenis ir atitinkamus failus, reikalingus modelių mokymui ir atpažinimui. Balso komandoms modeliuoti ir atpažinti taikant fonemų modelius reikia atlikti 8 nuoseklius duomenų ir failų paruošimo žingsnius. Pirmieji 7 žingsniai – fonemų PMM modelių paruošimas ir mokymas, o aštuntas žingsnis –

modelių testavimas. Taip pat vietoj būsenų ir Gauso mišinių variacijos įvedamos naujos fonemos. Akustinio modelio struktūra pateikta 7.2 pav.

Pirmiausia garsyno įrašai buvo transformuoti į požymių vektorių sekas. Tuo tikslu garso įrašai buvo diskretizuoti 16 kHz dažniu ir suskaidyti į 20 ms trukmės analizės langus, 10 ms paslinktus vienas kito atžvilgiu (persidengiantys analizės langai). Kiekviename analizės lange buvo įvertinama šnekos signalo energija ir signalo spektras. Spektro reikšmės buvo grupuojamos su 26 „filtrais“, kurie buvo išdėstyti netiesinėje (melų) dažnio skalėje. Remiantis filtrų išėjimais buvo apskaičiuota 12 melų dažnio kepstro koeficientų (MFCC). Signalo energijai ir kepstro koeficientams buvo papildomai apskaičiuojami jų pirmosios ir antrosios eilės skirtumai laiko atžvilgiu. Vieną 20 ms trukmės signalo analizės langą atitiko 39 komponentes turintis požymių vektorius.



7.2. pav. Fonemomis grįsto PMM akustinių modelių sudarymo procesas

Nelietuviško atpažintuvo naudojimas paremtas daugiakalbio atpažinimo principais, t. y. tikimasi, kad vienos kalbos (paprastai mažiau populiarios) fonetines savybes gana gerai atspindi kitos kalbos (paprastai populiarios) akustiniai-fonetiniai modeliai. Eksperimentai parodė (147), kad ispanų kalbos fonetinė sistema kur kas artimesnė lietuviškai nei angliška, todėl pasirinkta prie lietuvių kalbos adaptuoti

ispanų šnekos atpažintuvą (REC_SP), platinamą su *Windows'7* operacine sistema (146).

Microsoft balso serveris (*Microsoft Speech Server*, MSS'2007) yra interaktyvus telefoninis autoatsakiklis, integruotas kartu su paketu „Visual Studio 2005". Viena iš MSS'2007 ypatybių yra VoIP palaikymas. VoIP iš esmės leidžia vartotojams pateikti ir priimti užklausas per internetą. Balso serveris gali priimti VoIP užklausas be jokios papildomos programinės ar aparatinės įrangos. MSS'2007 palaiko anglų, vokiečių, ispanų, prancūzų, japonų ir kinų kalbas. MSS'2007 balso serveryje naudojama UPS (*Universal Phone Set*) transkribavimo sistema. Balso serverio atpažintuvas (REC_MSS) buvo naudojamas skaičiams atpažinti per telefoną.

7.2.2. Garsynai

Norint ištirti balso komandų atpažinimo tikslumą pirmiausia reikia turėti tinkamai paruoštus lietuviškų skaičių pavadinimų ir vardų balso komandų garsynus. Tyrimuose naudotų garsynų duomenys pateikti 7.1 lent.

Lietuviškų skaičių nuo 0 iki 9 pavadinimų garsyną SKAIC30 sudaro 30 diktorių – 23 moterų (M) ir 7 vyrų (V) – balso įrašai. Kiekvienas skaičius ištariamas 20 kartų. Tiriamasis lietuviškų skaičių pavadinimų garsynas sudarytas iš 6000 skirtingų balso įrašų.

7.1. lentelė. Garsynų duomenys

Garsyno pavadinimas	Komandų skaičius	Diktorių skaičius	Ištarimų skaičius
SKAIC30	10	30 (7 V, 23 M)	20
LETTERS	26	2 (1 V, 1 M)	50
NATO	26	2 (1 V, 1 M)	50
NAMES1	250	2 (1 V, 1 M)	20
NAMES2	70	10 (5 V, 5 M)	20
NAMES3	26	21 (9 V, 12 M)	20
LIEPA Z001	10	50 (9 V, 41 M)	1
LIEPA Z060	26 (18 frazių; 8 izoliuotos komandos)	143 (35 V, 108 M)	1
MEDIC	731	12 (7 V, 5 M)	20

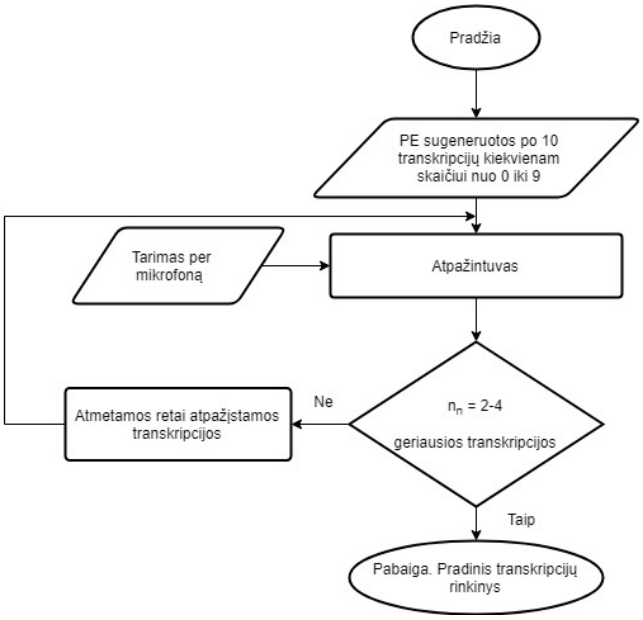
Garsyno LETTERS atpažinimo naudojant REC_SP atpažintuvą vidutinis tikslumas buvo tik 25,9 %. Vadinasi, raidės tarimas negali būti naudojamas kodams atpažinti. Garsynas NATO, su tuo pačiu atpažintuvu atpažintas 67,2 % tikslumu (157), taip pat negali būti naudojamas kodams atpažinti. Todėl buvo nuspręsta sukurti garsyną NAMES1. Jį sudaro apie 10 vardų, prasidedančių skirtingomis lotyniškos abėcėlės raidėmis. Šis garsynas buvo naudojamas pradiniame vardų atrankos etape (procedūra aprašyta 7.2.4 poskyryje). Geriausiai atpažinti 1, 2 ar 3 vardai ir kitokie žodžiai kiekvienai raidei pateko į NAMES2 garsyną. Šį papildė 8 diktorių įrašai. Galutinis vardų garsynas pavadintas NAMES3. Jį sudaro 21 diktoriaus – 12 moterų ir 9 vyrų – balso įrašai. Vardų garsyną sudaro 26 skirtingų vardų ir balso komandų

atitikmenys kiekvienai lotyniškos abėcėlės raidei. Kiekvienas vardas buvo ištartas 20 kartų ir garsyną sudaro iš viso 10 920 balso įrašų. Šis garsynas užtriukšmintas 5 dB lygyje ir panaudotas papildomam tyrimui su skirtingais atpažintuvais.

2015 m. rugpjūtį užbaigtas projektas „Lietuvių šneka valdomos paslaugos – LIEPA“. Projekto metu sukurtas 100 valandų garsynas buvo pritaikytas šnekos technologijų moksliniams tyrimams ir konstravimo darbams, elektroninėms paslaugoms teikti. Garsyno dalis Z001 buvo naudojama izoliuotoms komandoms atpažinti, o kita dalis Z060 – frazėms atpažinti. Garsyno dalis Z060 buvo pasirinkta tyrimui, nes joje yra didelis kalbėtojų skaičius ir daugiau negu trečdalis įrašų – frazės.

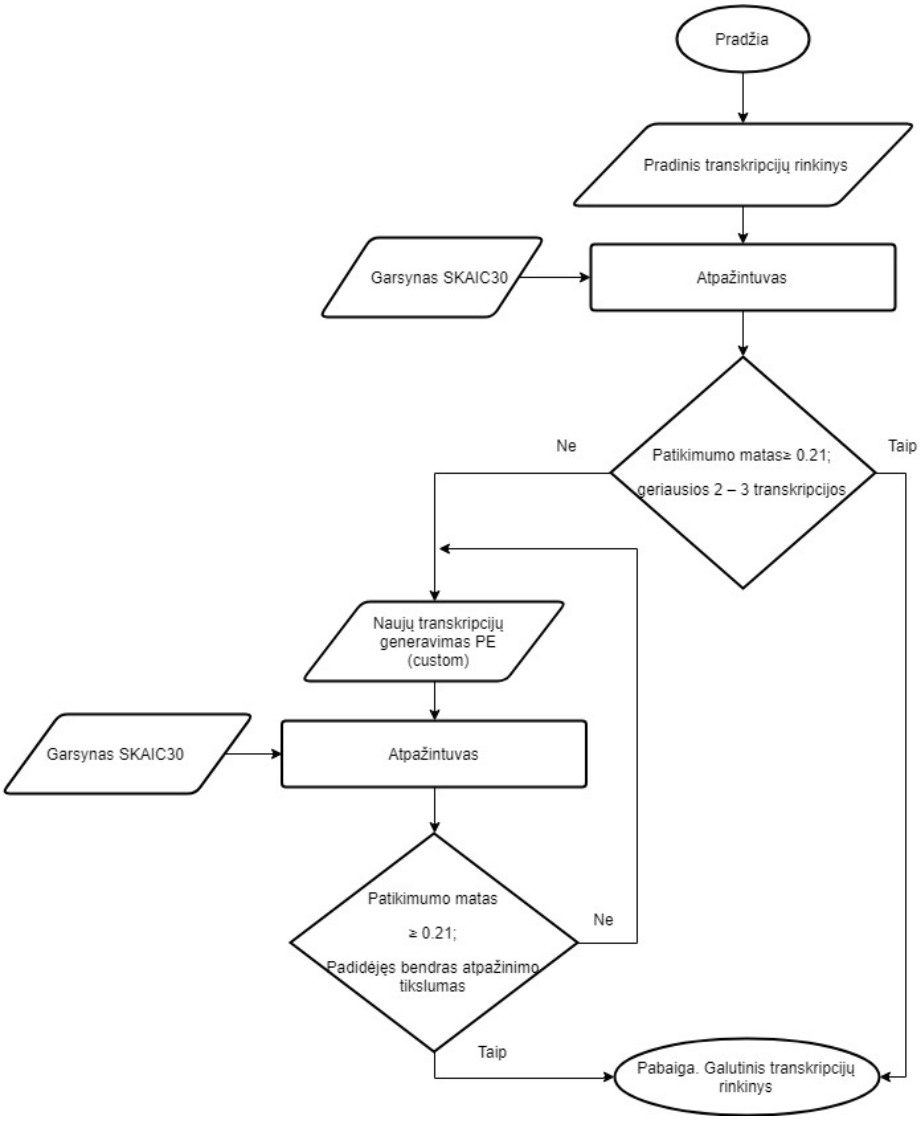
7.2.3. Lietuviškų balso komandų transkripcijų sudarymo metodika

Naudojant vokiečių, anglų, prancūzų bei ispanų kalbų sintezatorių ir atitinkamos kalbos UPS alfabetus sukurtos lietuviškų skaičių pavadinimų transkripcijos. Skaičių pavadinimai buvo sintezuojami, atrinkti tie, kurie skamba panašiausiai į lietuvišką tarimą. Sukurtos transkripcijos buvo nusiųstos *Microsoft* balso serverio (MSS’2007) gramatikos redaktoriui (*PE*). Kiekvienam skaičiui ir skirtingai kalbai parengti atskiri atrankos testai (iš viso 40). Bandymas buvo atliekamas *Microsoft* balso serverio paketu (MSS’2007), vienas diktorius ir viena diktorė kiekvieną skaičių per mikrofoną ištarė po 100 kartų. Pradinio transkripcijų rinkinio rengimo algoritmas pateiktas 7.3 pav.



7.3. pav. Pradinio transkripcijų rinkinio rengimo algoritmas

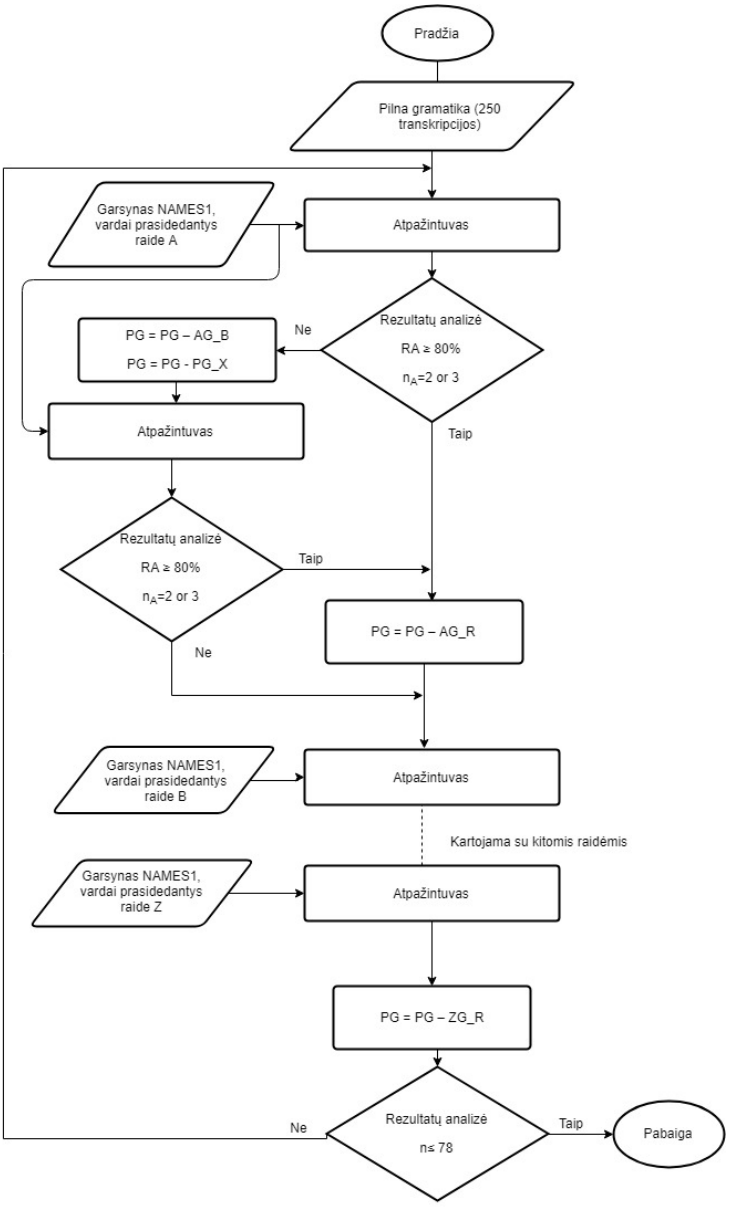
Daugiausia kartų atpažintos transkripcijos buvo naudojamos galutiniame transkripcijų paruošimo ir atrankos etape. Jo algoritmas pateiktas 7.4 pav.



7.4. pav. Galutinio transkripcijų rinkinio rengimo algoritmas

7.2.4. Vardų atrankos metodika

Vardų ir kitokių žodžių atitikmenų lotyniškai abėcėlei atranka buvo vykdoma keliais etapais, kurie pavaizduoti algoritmais 7.5 ir 7.6 pav. Atranka vykdyta naudojant adaptuotą ispanų šnekos atpažintuvą, esantį *Windows'7* operacinėje sistemoje.



7.5. pav. Pirminės vardų atrankos algoritmas

Pirminės atrankos iteracijos vykdomos pagal abėcėlę – pradedama nuo raidės „A“ ir pilnos gramatikos (*PG*).

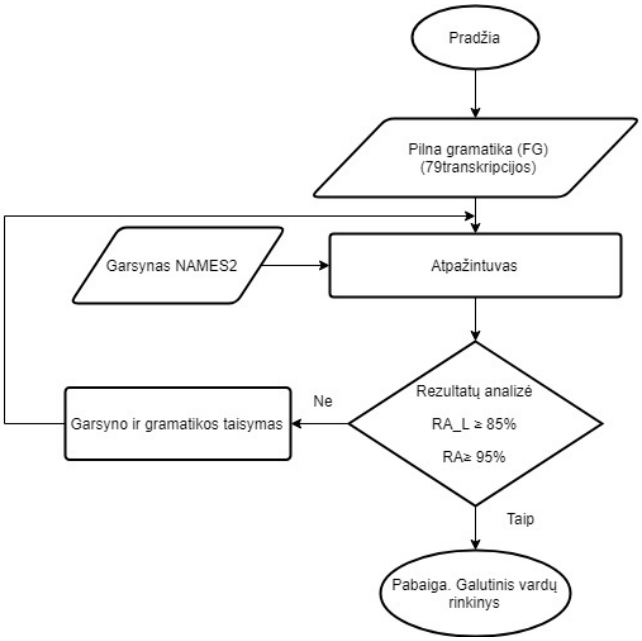
Kiekviename iteracijos žingsnyje vykdomas visų garsyne esančių vardų, prasidedančių testuojama raide, atpažinimo testavimas, skaičiuojami atpažinimo rezultatai ir atrenkami 1, 2 arba 3 geriausiai atpažinti vardai, jei jų atpažinimo tikslumas bent 80 %. Atpažinimo gramatikoje paliekamos tik atrinktų vardų transkripcijos, o kitos transkripcijos iš gramatikos pašalinamos (*AG_R*). Sudaroma testuojamos raidės vardų atpažinimo tikslumo eiliškumo lentelė.

Jeigu antrame žingsnyje nepavyko surasti nė vieno vardo, atpažįstamo bent 80 % tikslumu, visi vardai, prasidedantys testuojama raide, paliekami kitai iteracijai (raidės „A“ atveju – *AG_B*).

Tuo atveju, kai geriausiai atpažįstamo vardo atpažinimo tikslumas nesiekia 80 % ir iš atpažinimo rezultatų matyti, kad vardas yra maišomas su vardu, prasidedančiu kita raide, leidžiama iš gramatikos pašalinti testuojamą vardą atpažinti trukdantį vardą, jei tai nėra paskutinis likęs vardas, prasidedantis ta raide. Kritiniu atveju leidžiama pašalinti paskutinį vardą, prasidedantį kita raide, kuris trukdo atpažinti testuojamą vardą. Į atpažinimo gramatiką įtraukiamas kitas anksčiau pašalintas vardas pagal atpažinimo tikslumo eiliškumo lentelę. Kritinis atvejis – kai, pašalinus trukdantį vardą, žymiai padidėja testuojamo vardo atpažinimo tikslumas, o trukdantis vardas yra vienintelis atrinktas vardas, prasidedantis ta raide.

Kiekvienai raidei turi likti bent vienas vardas ir sąraše neturi likti 2 vardų su tais pačiais teksto fragmentais (*PG_X*), pavyzdžiui: Gražvydas, Mažvydas; Aleksas, Feliksas, Iksas.

Po pirminės atrankos sudaromas NAMES2 garsynas iš 70 vardų ir kitokių žodžių. Galutinės atrankos algoritmas pateiktas 7.6 pav. Atrenkami vardai ir kitokie žodžiai, atitinkantys lotyniškas raides.

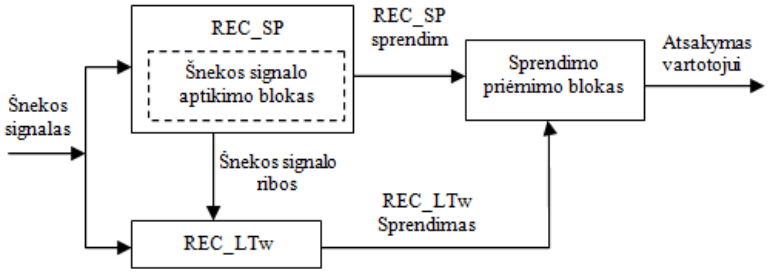


7.6. pav. Galutinės vardų atrankos algoritmas

Norint surinkti galutinį rinkinį, vieno vardo atpažinimo tikslumas (*RA_L*) turi būti didesnis nei 85 %, o bendras atpažinimo tikslumas (*RA*) turėtų būti didesnis nei 95 %.

7.2.5. Atpažintuvų sujungimo metodika

Hibridinis atpažintuvas sudarytas iš REC_SP atpažintuvo, turinčio integruotą signalo aptikimo bloką, REC_LTw atpažintuvo ir sprendimų priėmimo bloko, kuris realizuoja hibridinę (-es) sprendimo priėmimo taisyklę (-es) (7.7 pav.). Šnekos signalas pirmiausia patenka į REC_SP. Šis nustato komandos ribas ir signalo ištrauką pateikia REC_LTw atpažintuvui, o savo sprendimą perduoda sprendimų priėmimo blokui. REC_LTw atpažintuvas savo sprendimą taip pat perduoda sprendimų priėmimo blokui. Jei sprendimai skiriasi, sprendimų priėmimo blokas nusprendžia, kurį iš dviejų rezultatų pateikti vartotojui kaip galutinį atsakymą.



7.7. pav. Hibridinio atpažintuvo struktūra

Svarbiausias hibridinio atpažintuvo komponentas yra hibridinis sprendimų priėmimo blokas, kuris sukonstruotas taikant mašininio mokymo metodiką. Mokymui buvo naudojami garsyno įrašai, kai atpažintuvų REC_LTw ir REC_SP sprendimai skyrėsi. Jie glaustai apibūdinti 7.2 lent.

7.2. lentelė. Atpažintuvų REC_LTw ir REC_SP rezultatų papildomumas

Poaibis	Aprašymas
T=T	Atpažintuvų siūlomi sprendimai sutampa ir yra teisingi
F=F	Atpažintuvų siūlomi sprendimai sutampa ir yra neteisingi
T-	Atpažintuvas REC_LTw siūlo teisingą sprendimą, atpažintuvas REC_SP sprendimo nesiūlo
F-	Atpažintuvas REC_LTw siūlo neteisingą sprendimą, atpažintuvas REC_SP sprendimo nesiūlo
-T	Atpažintuvas REC_SP siūlo teisingą sprendimą, atpažintuvas REC_LTw sprendimo nesiūlo
-F	Atpažintuvas REC_SP siūlo neteisingą sprendimą, atpažintuvas REC_LTw sprendimo nesiūlo
--	Abu atpažintuvai sprendimo nesiūlo
TF	Atpažintuvų siūlomi sprendimai nesutampa, REC_LTw sprendimas teisingas
FT	Atpažintuvų siūlomi sprendimai nesutampa, REC_SP sprendimas teisingas
FF	Atpažintuvų siūlomi sprendimai nesutampa, abu sprendimai neteisingi

Kiekvieną mokymo imties objektą sudarė abiejų atpažintuvų sprendimai konkrečiam garso įrašui. Suformuluotas dviejų klasių – TF ir FT – atskyrimo (atpažinimo) uždavinys.

Požymiai, pagal kuriuos aprašoma mokymo imtis, hibridiniam sprendimų priėmimo blokui pateikti 7.3 lent. Kiekvienas mokymo imties objektas buvo aprašytas atsižvelgiant į 38 (skaičiams) ir 62 (vardams) požymius.

7.3. lentelė. Požymių imtis

Požymio pavadinimas	**Paaiškinimas**
SP_conf	REC_SP atpažintuvo pateikto sprendimo patikimumo įvertis [0,..,1000].
sp_supp	Jei REC_SP atpažintuvo pateiktas sprendimas sutampa su REC_LTw atpažintuvo pateikta 2-ąja (arba 3-iąja) alternatyva, šis parametras nurodo, kiek 2-oji (arba 3-ioji) REC_LTw alternatyva yra prastesnė už 1-ąjį (prioritetinį) REC_LTw sprendimą (logaritminės tikimybės prasme). Jei REC_LTw atpažintuvas nepateikia alternatyvių sprendimų arba jei REC_SP atpažintuvo siūlomas sprendimas nesutampa su REC_LTw atpažintuvo alternatyvomis, šiam požymiui priskiriama reikšmė 10.
lt_prob	REC_LTw atpažintuvo pateikto sprendimo patikimumo įvertis, matuojamas vidutine logaritmine tikimybe signalo analizės langui. Frazės pradžioje ir pabaigoje galimai esančios tylos atkarpos į šį įvertinimą neįtraukiamos.
lt_delta_prob	Patikimumo įverčių skirtumas tarp prioritetinio sprendimo ir 2-osios REC_LTw atpažintuvo alternatyvos. Jei REC_LTw atpažintuvas nepateikia alternatyvių sprendimų, šiam požymiui priskiriama reikšmė 10.
gender	Dvireikšmis požymis, nusakantis kalbėtojo lytį (m, f).
lt_a ..... lt_ž (*letters_lt*)	Proporcija (%), kurią REC_LTw atpažintuvo pateiktame prioritetiniame sprendime (frazėje) sudaro raidės „a“. Pvz., jei REC_LTw atpažintuvas pateikia prioritetinį sprendimą „AIDS“, tai šis požymis lygus 25 % (1 raidė iš 4).
sp_a ..... sp_ž *(letters_sp)*	Toliau tokiu pat būdu transformuojamas REC_SP atpažintuvo sprendimas, žr. prieš tai pateiktus požymių lt_a ..... lt_ž paaiškinimus.

Atpažintuvams sujungti taikomos dvi skirtingos metodikos:

1. Įprastinis 10 kartų kryžminis patikrinimas su grafine WEKA sąsaja. Paruošiamas vienas visų diktorių požymių failas, WEKA programų paketas pagal nutylėjimą atsitiktiniu būdu skirsto duomenis: 90 % mokymui, 10 % testavimui, 10 kartų atlieka klasifikavimą, tada rezultatus vidurkina ir parodo ekrane. Toks klasifikavimas leidžia prognozuoti klasifikavimo tikslumą (ir kartu hibridinio atpažintuvo tikslumą) žinomam kalbėtojui (vienam iš garsyno diktorių).

2. Sudėtingesnis n kartų kryžminis patikrinimas, kur n – diktorių skaičius. Paruošiama 2*n failų: mokymui imama n–1 diktorių, o testavimui – 1 (n-tojo) diktoriaus požymiai. Klasifikavimas atliekamas n kartų per komandinę eilutę paduodant mokymui n–1 diktorių požymių failą, o testavimui – n-tojo diktoriaus požymių failą. Tai kartojama n kartų keičiant n-tąjį diktorių. Tada rezultatai rankiniu

būdu vidurkinami. Tokio klasifikavimo rezultatai leidžia prognozuoti klasifikavimo tikslumą (ir kartu hibridinio atpažintuvo tikslumą) nežinomam kalbėtojui. Dėl didelės skaičiavimų apimties vietoje n kartų kryžminio patikrinimo daromas n/2, n/3 ir pan. kryžminis patikrinimas – gaunami mažesnio tikslumo rezultatai.

7.3. Atpažinimo tyrimai

7.3.1. Balso serveris REC_MSS

Tyrimai balso serveryje vykdyti su ispanų šnekos atpažintuvu *Microsoft Speech Recognizer 9.0 for MSS (Spanish-US)*. Naudojant lietuviškų skaičių pavadinimų garsyną SKAIC30 gautas 99,12±0,88 % atpažinimo tikslumas. Tyrimo rezultatai pateikti 7.4 lent.

7.4. lentelė. Skaičių pavadinimų atpažinimo tikslumo tyrimų naudojant balso serverį rezultatai

Skaičius	Atpažinimo tikslumas, %	Patikimumo rodiklis
NULIS	99,67±0,22	0,87
VIENAS	100	0,93
DU	98,83±0,42	0,90
TRYS	99,5±0,27	0,87
KETURI	95,5±0,80	0,74
PENKI	100	0,92
SESI	97,83±0,57	0,84
SEPTYNI	99,83±0,16	0,90
ASTUONI	100	0,86
DEVYNI	100	0,87
Vidurkis (%) su 95 % patikimumo intervalu	**99,12±0,88**	0,84

Tyrime taip pat buvo įvertintas patikimumo rodiklis (angl. *confidence measure*). Patikimumo rodiklis pateikiamas nuo 0 iki 1. Skaičius laikomas atpažintu, jei patikimumo rodiklis didesnis už 0,2.

7.3.2. Ispaniškas atpažintuvas REC_SP

Eksperimentai parodė, kad ispanų kalbos fonetinė sistema kur kas artimesnė lietuviškai nei kitos, todėl pasirinkta prie lietuvių kalbos adaptuoti ispanų šnekos atpažintuvą *Microsoft Speech Recognizer 8.0 (Spanish-US)*, platinamą su *Windows'7* operacine sistema.

Skaičių atpažinimas vykdytas su garsynu SKAIC30. Atpažinimo rezultatai su skirtingomis transkripcijų gramatikomis (UPS, žodinė ir maišyta), vyrišku ir moteriškų profiliais pateikti 7.5 lent.

7.5. lentelė. Skaičių atpažinimo su ispanišku atpažintuvu REC_SP tikslumo tyrimo rezultatai

Skaičius	Profilis ir gramatika						
	Default, žodinės	*Default*, UPS	Mot., žodinės	Mot., UPS	Vyr., žodinės	Vyr., UPS	Vyr., maišytos
NULIS	42,33	61,00	55,00	54,50	70,67	75,67	80,67±1,53
VIENAS	91,17	93,67	93,33	93,33	96,67	96,33	95,83±0,78
DU	64,33	67,00	51,67	51,33	73,50	80,00	79,67±1,56
TRYS	98,00	98,17	96,50	96,50	99,00	99,17	99,00±0,39
KETURI	53,83	57,83	49,50	48,83	85,83	74,50	86,50±1,33
PENKI	97,17	95,67	90,67	90,83	98,17	98,67	98,67±0,44
SESI	97,33	100	98,83	98,83	100	100	100
SEPTYNI	97,67	98,00	96,17	96,00	99,17	99,50	99,17±0,35
ASTUONI	95,33	95,67	86,50	87,17	99,67	99,67	99,67±0,22
DEVYNI	75,50	78,00	80,67	80,67	72,00	86,50	81,33±1,51
Vidurkis (%) su 95 % patikimumo intervalu	81,26± 12,99	84,81± 10,60	79,89± 12,36	79,79± 12,52	89,48± 7,88	91,01± 6,63	92,05± 5,48

Didžiausias skaičių atpažinimo tikslumo vidurkis gautas su maišytomis transkripcijomis ir vyrišku profiliu – 92,05±5,48 %.

Vardams atpažinti su REC_SP atpažintuvu buvo naudojamas NAMES3 garsynas. Atpažinimo rezultatai pateikti 7.6 lent.

7.6. lentelė. Vardų atpažinimo su ispanišku atpažintuvu REC_SP tikslumo tyrimo rezultatai

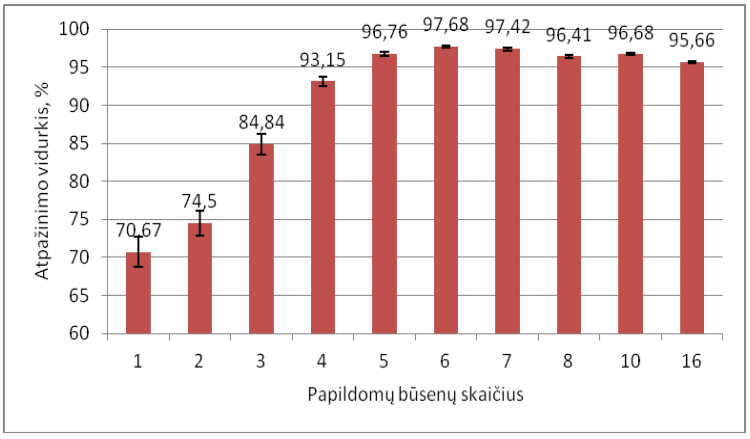
Vardas	Profilis ir gramatika				
	Mot., žodinės	*Default*, žodinės	Vyr., žodinės	Vyr., UPS	Vyr., maišytos
Austėja	90,8	98,7	99,6	99,2	99,6±0,29
Boleslovas	96,0	97,7	98,9	98,3	98,9±0,48
Cecilija	96,3	97,1	99,8	99,3	99,8±0,20
Donatas	98,3	98,7	99,8	99,8	99,8±0,20
Eimantas	98,9	98,9	97,9	97,9	97,9±0,66
Faustas	98,7	98,9	98,3	98,7	98,3±0,60
Gražvydas	95,0	98,9	99,4	98,7	99,4±0,36
Hansas	98,9	98,9	99,0	100,0	99,0±0,46
Izaokas	99,4	98,5	98,3	99,2	98,1±0,63
Jonas	94,0	96,3	97,1	96	97,1±0,78
Karolis	100,0	100,0	97,9	97,7	97,9±0,66
Laima	97,9	99,4	98,9	99	98,9±0,48
Martynas	99,6	98,3	97,7	99,8	97,1±0,78
Nojus	97,7	98,1	97,1	97,9	97,1±0,78
Oskaras	99,0	100	100,0	99,4	100
Patrikas	99,0	99,8	99,8	99,8	99,8±0,20
Qju	60,1	70,8	86,0	88,1	85,6±1,63

Ričardas	90,1	93,5	87,9	96,2	91,7±1,28
Sandra	97,7	99,8	99,4	99,8	99,2±0,41
Teodoras	95,6	99,8	97,1	97,9	97,1±0,78
Ulijona	92,3	94,8	96,0	90,4	96,0±0,91
Vacys	97,7	97,3	96,7	100	98,3±0,60
Wašington	93,3	96,9	98,5	97,88	98,5±0,56
Xsas	97,5	95,0	94,6	98,9	95,4±0,97
Ygrekas	93,1	95,6	96,0	96,2	95,4±0,97
Zacharijus	94,4	93,1	96,0	29,8	96,0±0,91
Vidurkis (%) su 95 % patikimumo intervalu	95,05±2,94	96,72±2,17	97,22±1,29	95,23±4,77	97,38±1,17

Kaip ir skaičių pavadinimų atpažinimo tyrime, didžiausias vardų atpažinimo tikslumas gautas su vyrišku profiliu ir maišytomis transkripcijomis – 97,38±1,17 %.

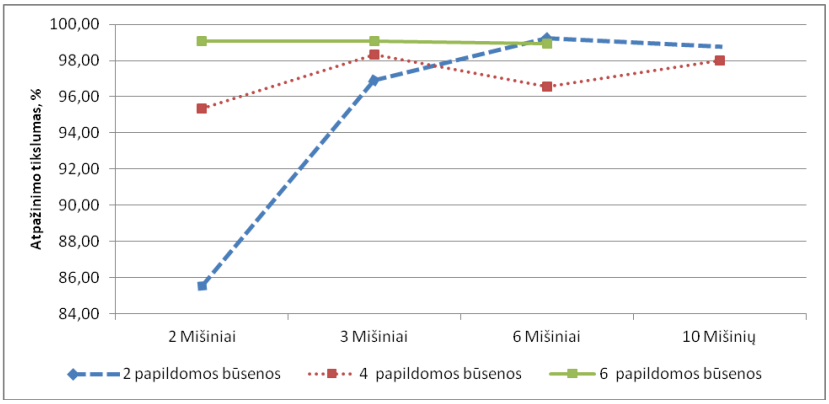
7.3.3. Lietuviškas atpažintuvas REC_LTw

Skaičių pavadinimų atpažinimo tikslumo tyrimui su skirtingu būsenų skaičiumi buvo parinktas būsenų skaičius, apytiksliai lygus balso komandą sudarančių fonetinių elementų skaičiui intervale imtinai nuo 2 iki 7 su pridėtu vienetu, dvejetu, ketvertu, penketu ir t. t. Atpažinimo rezultatai pateikti 7.8 pav.



7.8. pav. Skaičių pavadinimų atpažinimo tyrimų su papildomomis būsenomis rezultatai

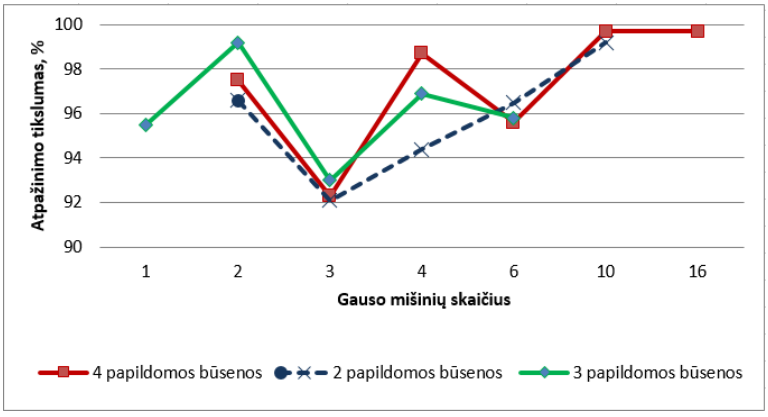
Toliau buvo tęsiami skaičių atpažinimo tikslumo tyrimai su skirtingu būsenų ir Gauso mišinių skaičiumi. Testuojant tą patį garsyną SKAIC30 buvo įterpiami Gauso mišiniai ir tiriamas komandų atpažinimo taikant modelius su papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi tikslumas. Atpažinimo rezultatai pateikti 7.9 pav.



7.9. pav. Skaičių atpažinimo tikslumo tyrimų su papildomomis būsenomis ir Gauso mišiniais rezultatai

Įvedus Gauso mišinius atpažinimo tikslumas žymiai padidėjo. Geriausi skaičių atpažinimo su REC_LTw atpažintuvu rezultatai (99,33±0,67 %) gauti naudojant 2 papildomas būsenas ir 6 Gauso mišinius. Su šiais akustinių modelių parametrais buvo atliktas 5 kartų kryžminis patikrinimas (mokoma su 24 diktorių įrašais ir testuojama su likusiais 6, kaskart keičiant mokymo ir testavimo diktorių rinkinius), gautas atpažinimo tikslumo vidurkis – 99,19±0,81 %.

Atitinkami tyrimai akustiniuose modeliuose keičiant būsenų ir Gauso mišinių skaičių buvo atlikti su vardų garsynu NAMES3. Atpažinimo tikslumo tyrimo rezultatai pateikti 7.10 pav.



7.10. pav. Vardų atpažinimo tikslumo tyrimo keičiant būsenų ir Gauso mišinių skaičių rezultatai

Geriausi vardų atpažinimo tikslumo tyrimo rezultatai (99,17±0,83 %) gauti su 3 papildomomis būsenomis ir 2 Gauso mišiniais. Su šiais akustinių modelių parametrais buvo atliktas 7 kartų kryžminis patikrinimas (mokoma su 19 diktorių įrašais ir testuojama su likusiais 3, kaskart keičiant mokymo ir testavimo diktorių rinkinius), gautas atpažinimo tikslumo vidurkis – 96,7±2,45 %.

7.3.4. Lietuviškas atpažintuvas REC_LTp

Šnekamosios kalbos atpažinimui taikyti HTK programinių įrankių paketą galima ir modeliuojant balso komandų atpažinimą fonemų modelių metodu. Šis balso komandų modeliavimo metodas buvo taikomas tik skaičių pavadinimų garsynui atsižvelgiant į tai, kad balsu ištartų kodų sistemai atpažinti reikės daugiausia skaičių (nes daugumoje kodų vyrauja skaičiai).

Tyrimui buvo sukurti 24 skirtingi fonemų rinkiniai, juos sudarė nuo 19 iki 35 fonemų. Kaip ir ankstesniuose tyrimuose, 24 diktorių įrašai buvo naudojami mokymo procesui, kiti 6 – testavimui. Tyrimų rezultatų dalis pavaizduota 7.7 lent.

7.7. lentelė. Skaičių pavadinimų garsyno atpažinimo tikslumo tyrimas taikant fonemomis grįstus PMM

Fonemų rinkinys	*Digit1*	*Digit2*	*Digit9*	*Digit16*
Fonemų skaičius rinkinyje	19	28 (SAMPA)	31(2)	35
Atpažinimo tikslumo vidurkis (5 kartų kryžminio patikrinimo), %	63,05±3,59	84,12±2,44	91,65±2,94	97,1±1,11

Geriausi tyrimo taikant fonemomis grįstus PMM rezultatai gauti su *Digit16* fonemų rinkiniu, kurį sudaro šios fonemos ir jų grupės: *vm, n, a, s, d, u, tm, rm, y, km, e, i, pm, shm, uo, lm, sil, t, ii, nk, sh, nm, dm, sm, ik, uk, ud, ish, esh, ek, en, et, ir, ir, sp*. Atlikus 5 kartų kryžminį patikrinimą gautas 97,1±1,11 % atpažinimo tikslumo vidurkis.

7.4. Hibridiškumo tyrimai

Atlikus literatūros analizę, dviejų atpažintuvų sujungimo galimybių tyrimui pasirinkta naudoti duomenų analizės sistemą WEKA, kurioje įdiegta kelios dešimtys klasifikatorių. Iš jų atrankos tyrimui pasirinkta naudoti: kNN (*K-Nearest Neighbour*), RIPPER, NB (*Naive Bayes*), RF (*Random Forest*), C4.5, *ZeroR*, SVM (*Support Vector Machines*), *AdaBoost*, MP (*Multilayer Perceptron*) ir MLR (*Multinomial Logistic Regression*) (159).

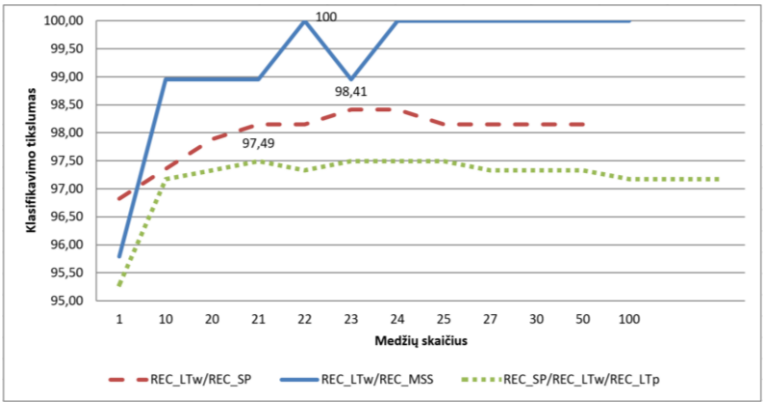
Atpažintuvų REC_LTw/REC_SP, REC_LTw/REC_MSS, REC_SP/ REC_LTw/REC_LTp sujungimo galimybių tyrimas atliktas naudojant skaičių pavadinimų garsyną. Hibridinė sprendimo priėmimo taisyklė buvo mokoma ir tikrinama 5 kartų kryžminio patikrinimo būdu ir įprastu 10 kartų kryžminio patikrinimo būdu. Šio eksperimento rezultatai pateikti 7.8 lent.

7.8. lentelė. Atpažintuvų sujungimo galimybių tyrimo naudojant skaičių pavadinimų garsyną rezultatai

Klasifikatorius	REC_LTw/REC_SP		REC_LTw/REC_MSS		REC_SP/ REC_LTw/REC_LTp	
	10 k. kryžminio patikrinimo vidurkis	5 k. kryžminio patikrinimo vidurkis	10 k. kryžminio patikrinimo vidurkis	5 k. kryžminio patikrinimo vidurkis	10 k. kryžminio patikrinimo vidurkis	5 k. kryžminio patikrinimo vidurkis
RIPPER	94,71	95,42	95,79	89,58	95,91	89,25
C4.5	96,83	95,31	90,53	78,16	99,02	90,69
MLR	97,09	96,04	97,89	81,55	93,45	85,39
MP	96,83	96,88	98,95	81,25	97,71	90,32
ZeroR	88,62	88,22	53,68	45,46	51,39	42,21
AdaBoost	97,35	96,95	95,79	92,08	93,78	90,37
kNN	96,56	94,77	96,84	89,88	96,07	86,67
RF	98,15	98,26	100	93,33	99,02	95,26
SVM	95,50	93,45	94,74	80,29	96,07	87,07
NB	92,06	85,83	93,68	80,16	85,11	83,81

Geriausi klasifikavimo rezultatai visais trimis atvejais gauti naudojant RF klasifikatorių (medžių skaičius 100). Geriausi klasifikavimo rezultatai gauti sujungus REC_LTw ir REC_SP atpažintuvus. Klasifikatoriaus išmokta sprendimo priėmimo taisyklių aibė veikia 99,02 % tikslumu ir hibridinio atpažintuvo veikimo tikslumas siekia 99,79±0,07 %, kai atliktas 10 kartų kryžminis patikrinimas.

Klasifikatorius suteikia galimybę keisti medžių skaičių, todėl buvo nuspręsta atlikti RF klasifikatoriaus efektyviausio medžių skaičiaus paieškos tyrimą. Tam buvo atliktas 10 kartų kryžminio patikrinimo eksperimentas keičiant RF klasifikatoriaus medžių skaičių nuo 1 iki 100. Tyrimo rezultatai pateikti 7.11 pav.



7.11. pav. Klasifikavimo tikslumo priklausomybė nuo medžių skaičiaus (naudojant skaičių pavadinimų garsyną)

7.9 lent. pateikti REC_LTw ir REC_LTp atpažintuvų (skaičių pavadinimų garsynui) sujungimo galimybių tyrimo su skirtingais klasifikatoriais rezultatai.

7.9. lentelė. Atpažintuvų sujungimo galimybių tyrimo naudojant skaičių pavadinimų garsyną rezultatai

Klasifikatorius	REC_LTw/REC_LTp	
	10 kartų kryžminio patikrinimo vidurkis	5 kartų kryžminio patikrinimo vidurkis
RIPPER	96,62	96,71
C4.5	97,97	97,31
MLR	99,32	99,13
MP	98,65	90,24
ZeroR	93,24	89,83
AdaBoost	99,32	96,71
kNN	100	99,13
RF	99,32	99,13
SVM	100	99,13
NB	100	97,31

Geriausi klasifikavimo sujungiant REC_LTw ir REC_LTp atpažintuvus rezultatai gauti naudojant kNN ir SVM klasifikatorius.

Dviejų atpažintuvų REC_LTw ir REC_SP sujungimo galimybių tyrimo naudojant vardų ir kitokių žodžių garsyną rezultatai pateikti 7.10 lent.

7.10. lentelė. Atpažintuvų sujungimo galimybių tyrimo naudojant vardų ir kitokių žodžių garsyną rezultatai

Klasifikatorius	REC_LTw/REC_SP	
	10 kartų kryžminio patikrinimo vidurkis	7 kartų kryžminio patikrinimo vidurkis
RIPPER	95,91	89,25
C4.5	99,02	90,69
MLR	93,45	85,39
MP	97,71	90,32
ZeroR	51,39	42,21
AdaBoost	93,78	90,37
kNN	96,07	86,67
RF	99,02	95,26
SVM	96,07	87,07
NB	85,11	83,81

10 kartų kryžminio patikrinimo eksperimentas parodė, kad RF klasifikatoriaus išmokta sprendimo priėmimo taisyklių aibė veikia 99,02 % tikslumu. Atsižvelgiant į tai, kad sprendimo taisyklė iškviečiama tik tada, kai REC_SP ir REC_LTw sprendimai skiriasi, vidutinis hibridinio atpažintuvo veikimo tikslumas siekia 99,44±0,09 %, kai atliktas 10 kartų kryžminis patikrinimas, ir 99,23 %, kai atliktas 5 kartų kryžminis patikrinimas.

Atliktas klasifikavimo tikslumo priklausomybės nuo medžių skaičiaus kitimo RF klasifikatoriuje tyrimas. Tikslumas svyravo: parinkus 46 medžius, tikslumas buvo 98,85 %, esant 48 medžiams, jis sumažėjo iki 98,19 %, o pasirinkus 100–150 medžių gautas stabilus 99,02 % hibridinio atpažintuvo veikimo tikslumas.

Norint patikrinti, ar sukurti akustiniai modeliai efektyvūs, atliktas sujungimo tyrimas su LIEPA garsynu. Pasirinkta dirbti su 50 diktorių (41 moters ir 9 vyrų) garsyno fragmentu. Skaičių nuo 0 iki 9 pavadinimų ištarimai buvo atpažinti per žodžiais grįstus ispanišką ir lietuvišką PMM atpažintuvus. Ispanų šnekos atpažintuvu REC_SP gautas 80,8±9,61 % atpažinimo tikslumas, o lietuvišku REC_LTw – 92±3,31 % atpažinimo tikslumas.

Atpažintuvai sujungti 10 kartų kryžminio patikrinimo būdu su 10 klasifikatorių (7.11 lent.).

7.11. lentelė. Atpažintuvų sujungimo galimybių tyrimo naudojant LIEPA garsyną rezultatai

Klasifikatorius	10 kartų kryžminio patikrinimo vidurkis
RIPPER	82
C4.5	82
MLR	70
MP	88
ZeroR	60
AdaBoost	82
kNN	88
RF	86
SVM	82
NB	82

Geriausi rezultatai gauti naudojant MLR ir KNN klasifikatorius. Abiem atvejais vidutinis hibridinio atpažintuvo veikimo tikslumas – 96,74±0,68 %. Palyginti su REC_LTw atpažintuvo rezultatais, klaidų sumažėjo 59,25 %. Tai leidžia patvirtinti, kad akustiniai modeliai ir sujungimo metodika veikia ir su kitais garsynais.

Frazių atpažinimo tyrimas taip pat buvo atliktas naudojant LIEPA šnekos garsyną. Garsyno dalį Z060 sudaro 143 kalbėtojų įrašai ir 26 komandos, iš kurių 18 frazių. Tyrimas buvo atliktas norint parodyti, kad šis metodas tinkamas ir frazėms atpažinti.

Naudojant *Kaldi* įrankių rinkinį (57), atpažinimo tyrimas buvo atliktas taikant monofoninius ir trifoninius akustinius modelius. Akustiniam modeliavimui naudoti tie patys MFCC požymiai, kaip ir dirbant su HTK programinių įrankių paketu, taip pat kiti numatytieji monofonų ir trifonų parametrai atpažinimui su *Kaldi* programinių įrankių paketu. Apie 20 % garsyno buvo naudojama testuoti, o likusi dalis – sistemai mokyti. Su visu garsynu Z060 buvo atliktas 5 kartų kryžminis patikrinimas, rezultatai suvidurkinti. Vidutinis frazių atpažinimo taikant monofoninius akustinius modelius tikslumas yra 86,04 %, o taikant trifoninius akustinius modelius – 89,95 %.

Sprendimų priėmimo taisyklę išmokusi sistema buvo išbandyta 10 kartų kryžminio patikrinimo metodu. Klasifikacijai buvo naudojami požymiai, gauti iš atpažintuvų išvesties, t. y. logaritminės tikimybės, o raidės proporcija žodyje ir diktoriaus lytis įvesti rankiniu būdu.

Geriausias klasifikacijos rezultatas buvo pasiektas su RF klasifikatoriumi. Pakeitus *Random seed* į *XVal/%Split* nuo 1 iki 40, vidutinis klasifikacijos taikant 10

kartų kryžminį patikrinimą tikslumas yra 93,07 % (standartinis nuokrypis 0,5). Kai atliekamas 10 kartų kryžminio patikrinimo bandymas, hibridinė sprendimo priėmimo taisyklė, išmokta RF klasifikatoriaus, veikia 93,44 ±0,15 % tikslumu. Palyginti su rezultatais, gautais naudojant tik trifoninio akustinio modelio atpažintuvą, klaidų sumažėjo 34,73 %.

Ankstesniuose tyrimuose duomenys naudoti netaikant papildomo signalo apdorojimo. Norint patikrinti, ar modelį būtų galima pritaikyti realiomis sąlygomis, tyrimui pasirinktas 5 dB lygyje užtriukšmintas vardų garsynas.

Naudojant *Kaldi* įrankių rinkinį, atpažinimo tyrimas buvo atliktas taikant trifoninį akustinį modelį. *Deep Speech 2* buvo pasirinktas kaip kitas atpažinimo variklis. *Deep Speech 2* yra *end-to-end* gilusis neuroninis tinklas, skirtas automatiniam šnekos atpažinimui, pagrįstas *Baidu* atpažinimo varikliu.

Naudojant anksčiau minėtus atpažintuvus buvo išmokytas NAMES3 šnekos garsynas (21 diktorius, 26 komandos, 20 ištarimų) ir atliktas 7 kartų kryžminis patikrinimas. 18 diktorių duomenys buvo naudojami mokymui, o 3 – testavimui.

Didesnis vidutinis atpažinimo tikslumas – 88,75 % – buvo pasiektas su *Kaldi* programinių įrankių paketu, taikant trifoninius akustinius modelius. Naudojant *Deep Speech 2* su RNN pasiektas tik 84,56 % atpažinimo tikslumas.

Atpažintuvams sujungti buvo naudojamas RF klasifikatorius. Pakeitus *Random seed* į *XVal/% Split* nuo 1 iki 40, vidutinis klasifikavimo taikant 10 kartų kryžminį patikrinimą tikslumas yra 92,62 % (standartinis nuokrypis 0,32). Hibridinė RF klasifikatoriaus išmokta sprendimų priėmimo taisyklė veikia 93,81±0,1 % tikslumu, kai taikomas 10 kartų kryžminio patikrinimo testas. Palyginti su rezultatais, gautais naudojant trifoninį atpažintuvą, klaidų sumažėjo 44,44 %.

7.5. Išvados

1. Surinktas ir paruoštas lietuviškų skaičių pavadinimų garsynas SKAIC30 (30 diktorių, 10 skaičių pavadinimų po 20 ištarimų) bei lietuviškų vardų ir kitokių žodžių garsynas NAMES3 (21 diktorius, 26 vardai po 20 ištarimų). Nustatyta, kad sukurta vardų ir kitokių žodžių atrankos metodika užtikrina didelį garsyno NAMES3 atpažinimo su ispanų šnekos atpažintuvu REC_SP tikslumą – 97,38±1,17 %. Palyginimui su tuo pačiu atpažintuvu buvo atliktas NATO alfabeto garsyno (2 diktoriai, 26 žodžiai po 50 ištarimų) atpažinimo tikslumo tyrimas ir gautas tik 67,2 % atpažinimo tikslumas.
2. Ispanų šnekos atpažintuvas *Microsoft Speech Recognizer 8.0* (*Spanish-US*) (REC_SP) buvo pasirinktas kaip kitakalbis atpažintuvas, o balso serveryje *Microsoft Speech Server* (MSS'2007) naudojamas ispanų šnekos atpažintuvas *Microsoft Speech Recognizer 9.0 for MSS (Spanish-US)* (REC_MSS) pasirinktas telefoniniam taikymui. Garsyno SKAIC30 atpažinimo su REC_MSS atpažintuvu tyrimai parodė, kad izoliuotų komandų transkripcijų atrankos metodika leidžia pasiekti didelį (99,12±0,88 %) skaičių pavadinimų garsyno atpažinimo su kitakalbiu atpažintuvu tikslumą.
3. Paruošti du lietuvių šnekos atpažintuvai. Jie ištirti taikant žodžiais ir fonemomis grįstus PMM. Pasiūlyta izoliuotų komandų atpažinimo metodika, kurią taikant pasirenkamas PMM būsenų ir Gauso mišinių skaičius žodžiais grįstuose PMM, leido pasiekti didelį garsyno SKAIC30 atpažinimo su žodžiais grįstu REC_LTw

atpažintuvu tikslumą (99,19±0,81 %). Pasiūlyta izoliuotų komandų atpažinimo įvedant naujus monofonus fonemomis grįstuose PMM metodika leido pasiekti pakankamai didelį garsyno SKAIC30 atpažinimo su fonemomis grįstu REC_LTp atpažintuvu tikslumą (97,1±1,11 %) net netaikant garsyno fonetinio segmentavimo.

4. Pasiūlyta kelių atpažintuvų sujungimo taikant mašininį mokymą metodika visais atvejais leido padidinti naudotų garsynų atpažinimo tikslumą:
   - sujungus du arba tris atpažintuvus, didžiausias skaičių pavadinimų garsyno atpažinimo tikslumas gautas su hibridiniu REC_LTw/REC_SP atpažintuvu (99,78 %); naudojant hibridinį REC_LTw/REC_SP atpažintuvą ir taikant 5 kartų kryžminio vidurkinimo metodiką klaidų sumažėjo labiausiai (72,84 %);
   - sujungus du atpažintuvus REC_LTw ir REC_MSS, gauti patys geriausi skaičių pavadinimų garsyno atpažinimo su hibridiniu atpažintuvu, skirtu telefoniniam signalui, rezultatai: taikant 10 kartų kryžminio vidurkinimo metodiką buvo gautas 100 % tikslumas, o taikant 5 kartų kryžminio vidurkinimo metodiką – 99,89 % tikslumas;
   - vardų garsyno NAMES3 atpažinimo su hibridiniu REC_LTw/REC_SP atpažintuvu taikant 10 kartų kryžminio vidurkinimo metodiką tikslumas lygus 99,44±0,09 %, o taikant 7 kartų kryžminio vidurkinimo metodiką – 99,23 %;
   - tyrimai su LIEPA šnekos garsynu parodė, kad sukurti akustiniai modeliai ir sujungimo metodika veikia tiek su kitais garsynais, tiek su frazėmis;
   - atliktas tyrimas su 5 dB lygyje užtriukšmintu vardų garsynu sujungiant su skirtingais atpažinimo varikliais (*Microsoft* ir *Baidu*) gautus garsyno atpažinimo rezultatus. Hibridinis atpažintuvas atpažinimo klaidų skaičių sumažino 44,44 %;
   - hibridinis atpažintuvas medicininio šnekos garsyno MEDIC atpažinimo klaidų skaičių sumažino iki 24 %, palyginti su vien HTK pagrindu veikiančiu lietuvišku atpažintuvu. Šio šnekos garsyno didžiąją dalį sudaro frazės (nuo 2 iki 5 žodžių).

5. Siūlomas atpažintuvų sujungimo būdas buvo taikomas hibridinio atpažinimo technologijoje, sukurtoje ir patvirtintoje projekto „INFOBALSAS“ metu.

6. Gauti rezultatai gali būti palyginti su įvairių Lietuvos autorių pasiektais rezultatais:
   - garsyno SKAIC30 atpažinimo su kitakalbiu atpažintuvu tikslumas yra gerokai didesnis nei nurodytas su R. Maskeliūno disertacijoje (83; p. 111) – 92,5 % (10 diktorių, 10 skaičių pavadinimų po 20 ištarimų);
   - lietuvių šnekos atpažintuvo REC_LTw tikslumas gali būti palygintas su S. Laurinčiukaitės disertacijoje (158; p. 78) pasiektu 97,77 % atpažinimo tikslumu (50 komandų, 31 diktorius, po 20 ištarimų);
   - rezultatai, gauti su lietuvių šnekos atpažintuvu REC_LTp, gali būti palyginti su S. Laurinčiukaitės disertacijoje (158; p. 78) pateiktu 93,91 % jau minėto 50 komandų garsyno atpažinimo tikslumu;
   - gauti hibridinių atpažintuvų garsynų atpažinimo tikslumo tyrimo rezultatai gali būti palyginti su T. Rasymo ir V. Rudžionio 2015 m. straipsnyje (131) pasiektu 50 komandų garsyno (12 diktorių, po 20 ištarimų) atpažinimo tikslumu (98,16 %), gautu sujungus penkis atpažintuvus (lietuvių, rusų, anglų, du vokiečių).

8. REFERENCES

1. HAWLEY, M, S., GREEN, P., ENDERBY, P., CUNNINGHAM, S., MOORE, R.K. Speech Technology for e-Inclusion of People with Physical Disabilities and Disordered Speech. *Interspeech*, Lisbon. 2005, pp. 445-448.
2. BENNACEF, S., DEVILLERS, L., ROSSET, S. and LAMEL, L. Dialog in the RAILTEL telephone-based system. *Spoken Language, ICSLP 96. Proceedings., Fourth International Conference on*, Philadelphia, PA. 1996, vol.1, pp. 550-553.
3. COX, R. V. et al. Speech and Language Processing for Next-Millenium. In *proceedings of the IEEE*. 2000, vol. 88, no. 8, pp. 1314-1337.
4. LARSON, J. A. Industry Perspectives and Business Opportunities. *ISCA Tutorial and Research & COST 278 Final Workshop: Applied Spoken Language Interaction in Distributed Environments (ASIDE 2005*, Denmark*)*. November 10-11, 2005.
5. The National Academy of Sciences. Voice Communication Between Humans and Machines. *National Academies Press*. 1994. pp. 321.
6. LEVITT H, A. Historical Perspective on Digital Hearing Aids: How Digital Technology Has Changed Modern Hearing Aids. *Trends in Amplification*. 2007, 11(1), pp. 7-24.
7. ZHOU, N., PFINGST, B.E. Relationship between multipulse integration and speech recognition with cochlear implants. *The Journal of the Acoustical Society of America*. 2014, 136(3), pp. 1257-1268.
8. GRATTAN, K.T.V., PALMER, A.W., SHURROCK, C.S.A. Speech recognition for the disabled Dept. of Electr., Electron. & Inf. Eng., City Univ., *London, Engineering in Medicine and Biology Magazine*. September 1991, vol. 10, no. 3, pp. 51-57.
9. DELLER, J. R., HANSEN, J. H. L., PROAKIS, J. G. Discrete-time processing of speech signals. *IEEE Press, Piscataway*. 2000. ISBN 0-7803-5386.
10. JUANG, B.H., LAWRENCE, R RABINER. Automatic speech recognition – a brief history of the technology development. Georgia Institute of Technology. *Atlanta Rutgers University and the University of California. Santa Barbara 1*. 2005.
11. FURUI, S. 50 years of Progress in speech and Speaker Recognition Research, *ECTI Transactions on Computer and Information Technology*. 2005, Vol.1, No. 2.
12. DAVIS, K. H., BIDDULPH, R., BALASHEK, S. Automatic Recognition of Spoken Digits. *The journal of the Acoustic society of America*. 1952, vol. 24, no. 6, pp. 627-642.
13. OLSON, H. F., BELAR, H. Phonetic Typewriter. *The journal of the Acoustic society of America*. 1956, vol. 28, no. 6, pp. 1072-1081.
14. FRY, D. B., DENES, P. The Design and Operation of the Mechanical Speech Recognizer at University College London. *The journal of British Inst. Radio Engr*. 1959, vol. 19, no. 4, pp. 211-229.
15. FORGIE, J. W., FORGIE, C. D. Results Obtained from a Vowel Recognition Computer Program. *The journal of the Acoustic society of America*. 1959, vol. 31, no. 11, pp. 1480-1489.
16. SUZUKI, J., NAKATA, K. Recognition of Japanese Vowels - Preliminary to the Recognition of Speech, *Journal Radio Research Laboratory*. 1961, vol. 37, no. 8, pp. 193-212.
17. SAKAI, J., DOSHITA, S. The Phonetic Typewriter. Information Processing. In *the proceedings of IFIP Congress*, Munich. 1962, pp. 445-450.
18. NAGATA, K., KATO, Y., CHIBA, S. Spoken Digit Recognizer for Japanese Language, *NEC Resource Development*. 1963, no. 6.
19. MARTIN, T. B., NELSON, A. L., ZADELL H. J. Speech Recognition by Feature Abstraction Techniques. *Tech. Report AL-TDR-64-176*, Air Force Avionics Lab. 1964.

20. VINTSYUK, T. K. Speech Discrimination by Dynamic Programming. *Kibernetika*. 1968, vol. 4, no. 2, pp. 81-88.
21. SAKOE, H., CHIBA, S. Dynamic Programming Algorithm Quantization for Spoken Word Recognition. *IEEE Trans. Acoustics, Speech and Signal Proc*. 1978, vol. ASSP-26, no. 1, pp. 43-49.
22. VITERBI, A. J. Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Trans. Information Theory*. April 1967, vol. IT-13, pp. 260-269.
23. ATAL, B. S., HANAUER S. L. Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. *The journal of the Acoustic society of America.* Aug. 1971, vol. 50, no. 2, pp. 637-655.
24. ITAKURA, F., SAITO S. A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies. *Electronics and Communications in Japan*. 1970, vol. 53A, pp. 36-43.
25. ITAKURA, F. Minimum Prediction Residual Principle Applied to Speech Recognition. In *IEEE Trans. Acoustics, Speech and Signal Proc*. Feb. 1975, vol. ASSP-23, pp. 57-72.
26. RABINER, L. R. et al. Speaker Independent Recognition of Isolated Words Using Clustering Techniques. In *IEEE Trans. Acoustics, Speech and Signal Proc*. 1979, vol. Assp-27, pp. 336-349.
27. REDDY, D. R. An approach to computer speech recognition by direct analysis of the speech wave. *Technical Report No. C549,* Computer Science Department, Stanford University, Stanford. 1966.
28. KLATT, D. Review of the ARPA speech understanding project. *J. Acoustic Soc. Am.* 1977, 62 (6), pp. 1324–1366.
29. LOWERRE, B. The HARPY Speech Understanding System, Trends in Speech Recognition. In *Readings in Speech Recognition*, Morgan Kaufmann Publishers. 1990, pp. 576-586.
30. ITAKURA, F. Minimum Prediction Residual Principle Applied to Speech Recognition. In *IEEE Trans. Acoustics, Speech and Signal Proc*. Feb. 1975, vol. ASSP-23, pp. 57-72.
31. KLATT, D. H. Review of the DARPA Speech Understanding Project. *The journal of the Acoustic society of America*. 1977, vol. 62, pp. 1345-1366.
32. TAPPERT, C.C., DIXON, N.R., RABINOWITZ, A.S., and CHAPMAN, W.D., Automatic Recognition of Continuous Speech Utilizing Dynamic Segmentation, Dual Classification, Sequential Decoding and Error Recover. *Rome Air Dev. Cen, Rome, NY, Tech. Report.* 1971, pp. 71-146.
33. JELINEK, F., BAHL, L.R., and MERCER, R.L., Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech, *IEEE Trans. Information Theory*. 1975, IT- 21, pp.250-256
34. JELINEK, F., The Development of an Experimental Discrete Dictation Recognizer. 1985, Proc.IEEE,73(11), pp.1616-1624.
35. RABINER, L.R., LEVINSON, S.E., ROSENBERG, A.E. and WILPON, J.G. Speaker Independent Recognition of Isolated Words Using Clustering Techniques. *Acoustics, Speech, Signal Proc.* August 1979, ASSP-27, pp. 336- 349.
36. FERGUSON, J. D. *Hidden Markov Analysis: An Introduction.* In: Ferguson, J. D. (ed.), Hidden Markov Models for Speech. IDA-CRD, Princeton, NJ. 1980.
37. CHOU, W. and JUANG, B. H., (Eds.) Pattern recognition in speech and language processing, *CRC Press*. 2003, pp. 115-147.
38. LIPPMANN, R.P. An introduction to computing with neural nets. IEEE Trans., *ASSP Mag*. 19874, (2), pp. 4-22,

39. LEE, K. F. et al. An overview of the SPHINX speech recognition system. *Proc. ICASSP*, 38. 1990, pp. 600-610.
40. JUANG, B.H., LEE, C.H. and WU, C. Minimum classification error rate methods for speech recognition. IEEE Trans. *Speech & Audio Processing*. 1997, T-SA, vo.5, No.3, pp.257-265.
41. VAPNIK, V. N. *Statistical Learning Theory*. John Wiley and Sons, 1998.
42. BROWN, M. K. et al. Advance in speech recognition technology. *IEEE Transaction on Signal Processing.* 1991, Vol.39, No.6.
43. YOUNG, S.J., JANSEN, J., ODELL, J.J., OLLASON, D., WOODLAND P.C. *The HTK Hidden Markov Model Toolkit Book*. Entropic Cambridge Research Laboratory, 1995.
44. AFIFY, M. and SIOHAN, O. Sequential estimation with optimal forgetting for robust speech recognition. *IEEE Transaction on Speech and Audio Processing*. 2004 Vol. 12, No.4.
45. RICCARDI, G. Activate learning: Theory and application to Automatic speech recognition. IEEE *Transaction on speech and Audio Processing*. 2005, Vol. 13, No. 4.
46. AFIFY, M., LIU, F., and JIANG, H. A new verification-based fast-match for large vocabulary continuous speech recognition. IEEE *Transaction on Speech and Audio Processing*. 2005, Vol. 133 No.4.
47. FURUI, S., TOMOHISA, I. et al. *Cluster-based Modeling for Ubiquitous Speech Recognition*. Department of Computer Science Tokyo Institute of Technology, Interspeech, 2005.
48. FURUI, S. Recent progress in corpus-based spontaneous speech recognition. *IEICE Trans. Inf. & Syst*. 2005, E88-D, 3, pp. 366-375.
49. KOO, M. W., LEE, C. H., JUANG, B. H. Speech recognition and utterance verification based on a generalized confidence score. IEEE *Trans. Speech Audio Process*. 2001, 9, pp. 821–832.
50. De WACHTER, M., MATTON, M., DEMUYNCK, K., WAMBACQ, P., COOLS, R. and COMPERNOLLE, D. Template-based continuous speech recognition. IEEE *transactions on audio, speech, and language processing.* 2007, vol. 15, no. 4, pp. 1377-1390.
51. SLOIN, A. et al., Support Vector Machine Training for improved hidden Markov modeling. *IEEE Transactions on Audio, Speech and Language processing.* 2008, Vol.56, No.1.
52. CUI, X. et al. A Study of Variable-Parameter Gaussian Mixture Hidden Markov Modeling for Noisy Speech Recognition. IEEE *Transactions On Audio, Speech, And Language Processing.* 2007, Vol. 15, No. 4.
53. HINTON, G. E., DENG, L., YU, D., DAHL, G. E., MOHAMED, A., JAITLY, N., SENIOR, A., VANHOUCKE, V., NGUYEN, P., SAINATH, T. and KINGSBURY, B. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine.* 2012, (November):82–97. Issue: 6, pp. 82-97.
54. GODFREY, J. and HOLLIMAN, E. Switchboard-1 Release 2 LDC97S62 [online]. Philadelphia*: Linguistic Data Consortium*, 1993. [Retrieved 2020-10-25]. Access: https://catalog.ldc.upenn.edu/LDC97S62.
55. CANAVAN, A., GRAFF, D. and ZIPPERLEN, G. CALLHOME American English Speech LDC97S42 [online]. Philadelphia*: Linguistic Data Consortium*, 1997. [Retrieved 2020-10-25]. Access: https://catalog.ldc.upenn.edu/LDC97S42.
56. ZHENG, J., FRANCO, H. and STOLCKE, A. Modeling word-level rate-of-speech variation in large vocabulary conversational speech recognition. *Speech Communication.* 2003, 41, pp.273-285.

57. POVEY, D., GHOSHAL, A., BOULIANNE, G., BURGET, L., GLEMBEK, O., VESEL, Y. N., GOEL, K., HANNEMANN, M., MOTLICEK, P., QIAN, Y., SCHWAR, Z P., SILOVSKY, J. and STEMMER, G. *The Kaldi speech recognition toolkit. In ASRU*. 2011.
58. GAIDA, C., LANGE, P., PETRICK, R., PROB, P., MALATAWY, A., SUENDERMANN-OEFT, D. *Comparing open-source speech recognition toolkits*. Technical report, DHBW, 2014.
59. RATH, P.S., POVEY, D., VESELY, K., CERNOCKY, J. Improved feature processing for deep neural networks. In: Proc. of Interspeech 2013, *International Speech Communication Association.* 2013, pp. 109–113.
60. GAIDA, C. et al., *Comparing Open-Source Speech Recognition Toolkits*. Tech. Rep., DHBW Stuttgart, 2014.
61. SAON, G., KUO, H.-K. J., RENNIE, S. and PICHENY, M. The IBM 2015 English conversational telephone speech recognition system. In Proc. *Interspeech*. 2015, pp. 3140– 3144.
62. SAON, G., SERCU, T., RENNIE, S. J. and KUO, H. J. The IBM 2016 English conversational telephone speech recognition system. In Proc. *Interspeech*. 2016, pp. 7–11.
63. XIONG, W., DROPPO, J., HUANG, X., SEIDE, F., SELTZER, M., STOLCKE, A., YU, D. and ZWEIG, G. The Microsoft 2016 conversational speech recognition system. In. *IEEE ICASSP - 2017*. 2017, pp.5255-5259.
64. SAON, G., KURATA, G., SERCU, T., AUDHKHASI, K., THOMAS, S., DIMITRIADIS, D., CUI, X., RAMABHADRAN, B., PICHENY, M., LIM, L., ROOMI, B., HALL, P. English Conversational Telephone Speech Recognition by Humans and Machines. *Proc. Interspeech* 2017. 2017, pp. 132-136.
65. XIONG, W., WU, L., ALLEVA, F., DROPPO, J., HUANG, X., & STOLCKE, A. The Microsoft 2017 Conversational Speech Recognition System. *CoRR, abs/1708.06073*. 2017.
66. DEMPSEY, P. The teardown Amazon echo digital personal assistant [Teardown Consumer Electronics]. *Engineering & Technology*. 2015, vol. 10, no. 2, pp. 88-89.
67. DEMPSEY, P. The teardown: Google Home personal assistant. *Engineering & Technology*. 2017, vol. 12, no. 3, pp. 80-81.
68. SARIKAYA, R., et al. An overview of end-to-end language understanding and dialog management for personal digital assistants. *2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA. 2016, pp. 391-397.
69. AMODEI, D., ANUBHAI, R., BATTENBERG, E., CASE C. et al. Deep speech 2: End-to-end speech recognition in English and mandarin. *arXiv preprint arXiv:1512.02595*. 2015.
70. BATTENBERG, E. et al. Exploring neural transducers for end-to-end speech recognition *2017 IEEE Automatic Speech Recognition and Understanding Workshop* (ASRU), Okinawa. 2017, pp. 206-213, doi: 10.1109/ASRU.2017.8268937.
71. LIPEIKA, A., LIPEIKIENĖ, J., TELKSNYS, L. Development of Isolated Word Speech Recognition System. *Informatica*. 2002, *13(1*), pp. 37-46.
72. TAMULEVIČIUS, G., LIPEIKA, A. Žodžių atpažinimo sistemos kūrimas. Lietuvos matematikos rinkinys (ISSN 0132-2818), T.43, spec. nr. *Lietuvos matematikų draugijos XLIV konferencijos mokslo darbai. 2003*, p. 292-296.
73. LAURINČIKAITĖ, S., Atskirai pasakytų lietuvių kalbos žodžių atpažinimas, remiantis paslėptais Markovo modeliais, *Informacinės technologijos.* Kaunas. Technologija. 2003, IX-21-24.
74. FILIPOVIČ, M. Atskirai pasakytų žodžių atpažinimo, naudojant neuroninius tinklus, tyrimas. *Informacinės technologijos*. Kaunas: Technologija. 2003, IX-10-20.

75. FILIPOVIČ, M. *Atskirai tariamų lietuvių šnekos žodžių atpažinimo, grindžiamo dirbtiniais neuroniniais tinklais ir paslėptais Markovo modeliais, tyrimai.* Ph.D. Thesis. Vytautas Magnus University. Institute of mathematics and informathics. 2005, Kaunas.
76. SKRIPKAUSKAS, M. Lietuvių šnekos signalų segmentavimas kvazifonemomis. *Informacinės technologijos.* Kaunas: Technologija. 2006, p.76 -80.
77. RAŠKINIS, G., RAŠKINIENĖ, D. Lietuvių šnekos atpažinimo sistemos, pagrįstos paslėptais Markovo modeliais, parametrų tyrimas ir optimizacija, *Informacinės technologijos*. Kaunas, Technologija. 2003, IX-41-48.
78. ŠILINGAS, D. *Akustinių lietuvių šnekos atpažinimo modelių parinkimas, naudojant paslėptus Markovo modelius*. Ph.D. Thesis. 2005, Kaunas.
79. RAŠKINIS, A., RAŠKINIS, G., KULIEŠIENĖ, D. Antros eilės požymių sistema šnekos signalo segmentavimo taškų atpažinimui. *Informacinės technologijos.* Kaunas: Technologija. 2005, p.294 -298.
80. RAŠKINIS, A., DEREŠKEVIČIŪTĖ, S. Dusliųjų sprogstamųjų priebalsių požymių tyrimai. *Informacinės technologijos*. Kaunas: Technologija. 2006, p.99 -103.
81. NOREIKA, S., RUDŽIONIS, A. Phoneme-like model of speech signal. In Proceedings of the XII *International Congress of Phonetic Sciences. Aix-En-Provence. France.* 1991, Vol.4, pp. 490-493.87.
82. RUDŽIONIS, A. Recognition by averaged templates. COST 249 Continuous Speech Recognition Over the Telephone, Draft Minutes of the 1st Management Committee Meeting. Brussel, Belgium. 1994, pp. 41-47.
83. MASKELIŪNAS, R. *Lithuanian Voice Commands Recognition Based on the Multiple Transcriptions*. Ph.D. Thesis, KTU, Kaunas: Technologija, 2009.
84. RUDŽIONIS, V. *Speech Recognition by phonetic units.* Ph.D. Thiesis, KTU, Kaunas: Technologija, 1998.
85. RUDŽIONIS, A., RUDŽIONIS, V. Phoneme recognition in fixed context using regularized discriminant analysis. Eurospeech'99 Proceedings, *ESCA 6th European Conference on Speech Communication and Technology*, ISSN 1018-4074. Budapest, Hungary. 1999 September 5-9, pp. 2745 – 2748.
86. DAUNYS, G., BALBONAS, D. Garsų klasifikavimas panaudojant sprendimų medžius. *Informacinės technologijos*. Kaunas: Technologija. 2005, p.277-282.
87. BALBONAS, D., DAUNYS G. Fonemų klasifikavimas panaudojant garso ir vaizdo informaciją. *Elektronika ir elektrotechnika.* ISSN 1392-1215. 2005, Nr.5(61), p.74-77.
88. RUDŽIONIS, V., RAŠKINIS, G., MASKELIŪNAS, R., RUDŽIONIS, A., RATKEVIČIUS, K., BARTIŠIŪTĖ, G. Web services based hybrid recognizer of Lithuanian voice commands. *Electronics and electrical engineering. Kaunas*: KTU. 2014, Vol. 20, no. 9, pp. 50-53.
89. RASYMAS, T., RUDŽIONIS, V. Lithuanian digits recognition by using hybrid approach by combining Lithuanian Google recognizer and some foreign language recognizers. *Communications in computer and information science*. 2015, vol. 538, pp. 449-459.
90. SIPAVIČIUS, D., MASKELIŪNAS, R. Google Lithuanian Speech Recognition Efficiency Evaluation Research. *Information and Software Technologies. ICIST 2016*. Communications in Computer and Information Science. 2016, vol. 639, pp. 602–612.
91. MENDELS, G., COOPER, E., SOTO, V., HIRSCHBERG, J., GALES, M., KNILL, K., RAGNI, A. and WANG, H., Improving speech recognition and keyword search for low resource languages using web data. *In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2015 - January, pp. 829–833.

92. DAVEL, M., BARNARD, E., VAN HEERDEN, C., HARTMANN, W., KARAKOS, D., SCHWARTZ, R. and TSAKALIDIS, S. Exploring minimal pronunciation modeling for low resource languages. *In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2015-January, pp. 538–542.
93. GALES, M. J. F., KNILL, K. M., RAGNI, A. Unicode-based graphemic systems for limited resource languages. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2015, pp. 5186- 5190.
94. ALUMAE, T., TILK, O. Automatic Speech Recognition System for Lithuanian Broadcast Audio. *Proceedings of the Seventh International Conference Baltic hlt 2016* "human language technologies – the Baltic perspective", IOS Press BV. 2016, vol. 289, pp. 39-45.
95. SALIMBAJEVS, A., KAPOCIUTE-DZIKIENE, J. General-Purpose Lithuanian Automatic Speech Recognition System. *In Proceedings of the 8th International Conference, Baltic HLT*, Tartu, Estonia. 2018, pp. 150–157.
96. GREIBUS, M., RINGELIENE, Ž., TELKSNYS, A. L. The phoneme set influence for Lithuanian speech commands recognition accuracy. *In Proceedings of the conference Electrical, electronic and information sciences* (eStream). Vilnius. 2017, pp. 1-4.
97. LILEIKYTE, R., LAMEL, L., GAUVAIN, J., GORIN, A. Conversational Telephone Speech Recognition for Lithuanian. *Computer Speech and Language*. 2018, vol. 49, pp. 71-92.
98. RAŠKINIS, G., PAŠKAUSKAITĖ, G., SAUDARGIENĖ, A., KAZLAUSKIENĖ, A., VAIČIŪNAS, A. Comparison of Phonemic and Graphemic Word to Sub-Word Unit Mappings for Lithuanian Phone-Level Speech Transcription. *Informatica.* 2019, Vol. 30, Issue 3, pp. 573–593.
99. PIPIRAS, L., MASKELIŪNAS, R., DAMAŠEVIČIUS, R. Lithuanian Speech Recognition Using Purely Phonetic Deep Learning. *Computers.* 2019, vol. 8, no. 76, pp.1-15.
100. RODMAN, D. R. *Computer Speech Technology*. Boston, Mass.: *Artech House*, 1999.
101. HUI-LING, L., *Toward a high-quality singing synthesizer with vocal texture control*. PhD Thesis, Stanford University, 2002.
102. RABINER, L., JUANG, B.H., *Fundamentals of speech recognition.* Prentice Hall. New Jersey, 1993.
103. JANKOWSKI, C.R., HOANG-DOAN, H., LIPPMANN, L.P. A Comparison of Signal Processing Front Ends for Automatic Word Recognition. *IEEE Transactions on Speech and Audio Processing. 1995,* Vol. 3, no. 4. pp. 286-292.
104. DAVIS, S.B., MERMELSTEIN, P. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1980, vol. ASSP-28, no. 4, pp. 357-366.
105. ATAL, B. S. and HANAUER, S. L. Speech analysis and synthesis by linear prediction of the speech wave, *Journal of the Acoustical Society of America*. 1971, vol. 50(2), pp. 637-655.
106. MAKHOUL, J. Linear Prediction: A Tutorial Review. Proc. of the IEEE. 1975, vol. 64.
107. DOMANTAS, A., RUDŽIONIS, A. Towards More Reliable Automatic Recognition of the Phonetic Units. In Proc. Of 12th Int. Congress of Phonetic Sciences, Aix-En-Provence. 1991, vol. 4.
108. HUANG, X., LEE, K.F., HIN, H.W., HWANG, M.Y. Improved Acoustic Modelling with the SPINX Speech Recognition System. In Proceedings ICASSP-91. 1991, Vol. 1.
109. PARKER, M. *Digital Signal Processing*. Chapter 10 – Discrete and fast Fourier transforms (DFT, FFT). 2010, pp. 97-112.
110. HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. Prentice –Hall, 1999.

111. OPPENHEIM, A.V., SCHAFER, W., STOCKHAM, T.G. Nonlinear filtering of multiplied and convolved signals. *Proceedings of the IEEE*. 1968, vol. 56. pp. 1264-1291.
112. YOUNG, S., EVERMANN, G., KERSHAW, D., MOORE, G., ODELL, J., OLLASON, D., VALTCHEV, V., WOODLAND, P. *The HTK Book.* Microsoft Corporation, 2001.
113. RABINE, L. R., *The HTK Book* and selected applications in speech recognition. *Proceedings of the IEEE*. Feb 1989, vol. 77, no. 2, pp. 257-286.
114. HUANG, X., ARIKI, Y. and JACK, M. A. Hidden Markov Models for Speech Recognition. Edinburgh Univ. Press, Edinburgh. 1990.
115. BAUM, L. E. An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process. *Inequalities*. 1972, vol. 3, pp. 1-8.
116. FURUI, S. *Digital speech processing, synthesis, and recognition*. New York, 1989.
117. SAKOE, H. and CHIBA, S. 'Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustic., Speech, Signal Processing*, 1978, ASSP-26, 1, pp. 43-49.
118. ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 6. 1958. pp. 386–408.
119. DZEMYDA, G., VEIKUTIS, V., JAKUŠKA, P., PUODŽIUKYNAS, A., TREIGYS, P., MEDVEDEV, V. Development of special data mining methods to explore the anisotropy of texture's temperatures of the heart. Report for project No. T-08153. Vinius, Institute of Informatics and Mathematics. 2008.
120. LIU, L., HAN, B., REN, X., GAO, Z. Learning algorithm for the state feedback artificial neural network, Sixth International Conference on Natural Computation *(ICNC 2010).* 2010, vol. 1, pp. 357–361.
121. SAON, G., Jen-TZUNG, C. Large-Vocabulary Continuous Speech Recognition Systems: A Look at Some Recent Advances. *In: IEEE Signal Processing Magazine*. 2012, vol. 29(6), pp. 18-33.
122. KUMAR, N., ANDREOU, A. Heteroscedastic Discriminant Analysis and Reduced Rank HMMs for Improved Speech Recognition. In*: Speech Communication*, 1998, vol. 25, no. 4, pp. 283-297.
123. BARTIŠIŪTĖ, G., RATKEVIČIUS, K., PAŠKAUSKAITĖ, G. Hybrid recognition technology for isolated voice commands. *Advances in intelligent systems and computing*, Springer. 2016 vol. 432, pp. 207-216.
124. TRENTIN, E., GORI, M. A survey of hybrid ANN/HMM models for automatic speech recognition. *Neurocomputing.* 2001, vol. 37, pp. 91-126.
125. HAFNER, P., WAIBEL, A., SHIKANO, K. Fast back-propagation learning methods for large phonemic neural networks. In Proceedings of *Eurospeech*. 1991.
126. TEBELSKIS, J., WAIBEL, A., PETEK, B., SCHMIDBAUER, O. Continuous Speech recognition using Linked Predictive Networks. Advances in Neural Information Processing Systems. *Denver. CO. Morgan Kaufman. San Mateo*. 1991, pp. 199-205.
127. KIMBER, D., BUSH, M.A., TAJCHMAN, G.N. Speaker – independent vowel classification using hidden Markov models and LVQ2. In Proceedings of the International Conference on Acoustics, *Speech and Signal Processing*. 1990, pp. 497-500.
128. RIGOLL, G. Maximum mutual information neural networks for hybrid connectionist – HMM speech recognition systems. *IEEE Transactions on Speech and Audio Processing.* 1994, vol. 2. no 1. pp. 175-184.
129. BENGIO, Y., GORI, M., De MORI, R. Learning the dynamic nature of speech with backpropagation for sequences. *Pattern Recognition letters.* 1992, 13(5), pp. 375-386.

130. SCHWENK, H. and GAUVAIN, J.-L. Combining multiple speech recognizers using voting and language model information. In IEEE international conference on spoken language processing *(ICSLP)*, II Pekin. 2000 pp. 915-918.
131. RASYMAS, T., RUDŽIONIS, V. Evaluation of Methods to Combine Different Speech Recognizers. *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, Lodz, Poland*. 2015, pp. 1043-1047.
132. RASYMAS, T., RUDŽIONIS, V. Combining different speech recognizers by using CART classifier. *2015 IEEE 3rd Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, Riga, Latvia. 2015, pp. 1-4, doi: 10.1109/AIEEE.2015.7367296.
133. RASYMAS, T., RUDŽIONIS, V. Lithuanian digits recognition by using hybrid approach by combining Lithuanian Google recognizer and some foreign language recognizers. *Information and sowtware technologies: 21st international conference ICIST 2015, Druskininkai*. Proceedings: Communications in computer and information science. 2015, vol. 538, p. 449-459.
134. JUDITH, A. M. *Using Speech Recognition*. Prentice Hall *PTR*, 1996, 292 p.
135. VARGA, A., STEENEKEN, H.J.M. Assessment for automatic speech recognition II: NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*. 1993, vol.12, No.3, pp. 247-252.
136. PEARCE, D., HIRSCH, H.-G. The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy condition. Sixth International Conference on Spoken Language Processing, *ICSLP 2000, INTERSPEECH 2000*, Beijing, China. 2000.
137. LEONARD, R.G. A database for speaker independent digit recognition. *ICASSP84*. 1984, vol.3, pp. 42.11.
138. RUDŽIONIS, A., RUDŽIONIS, V. Lithuanian speech database LTDIGITS. *Proceedings of LREC 2002*, Las Palmas, Spain. 2002, pp. 877-882.
139. RUDŽIONIS, A., RUDŽIONIS, V., ŽVINYS, P. Lietuvių kalbos signalų duomenų bazės LTDIGITS akustinės-fonetinės charakteristikos. *Baltų kalbų fonetikos ir akcentologijos problemos*, St. Peterburgas. 1999.
140. Lietuvių kalbos plėtros informacinėse technologijose 2014–2020 m. Gairės. Iš Valstybinės lietuvių kalbos komisijos 2013 m. spalio 24 d. posėdyje, protokolo Nr. P-7.
141. SCHIEL, F. and DRAXLER, C. The production of speech corpora. *Bavarian Archive for Speech Signals*, Tech. Rep., 2003.
142. FRAGA-SILVA, T., LAURENT, A., GAUVAIN, L., LAMEL, L., LE V., MESSAOUDI, A. Improving data selection for low-resource STT and KWS. 2015 *IEEE Workshop on Automatic Speech Recognition and Understanding* (ASRU). 2015, pp.153-159.
143. BESACIER, L., BARNARD, E., KARPOV, A., SCHULTZ, T. Automatic speech recognition for under-resourced languages. *A survey. Speech Communication*. 2014, vol. 56, p.85–100.
144. SAMPA – Speech Assessment Methods Phonetic Alphabet [online]. 2014. [Retrieved 2017-05-25]. Access: http://www.phon.ucl.ac.uk/home/sampa/index.html.
145. RAŠKINIS, A., RAŠKINIS, G., KAZLAUSKIENĖ, A. SAMPA (Speech Assessment Methods Phonetic Alphabet) for Encoding Transcriptions of Lithuanian Speech Corpora. *Information technology and control.* 2003, pp. 52-55.
146. MASKELIŪNAS, R., RUDŽIONIS, A., RATKEVIČIUS, K. Investigation of Foreign Languages Models for Lithuanian Speech Recognition. *Elektronika Ir Elektrotechnika (Electronics and Electrical Engineering)*. 2009, no. 3(91), pp.15-20.

147. BARTISIUTE, G., RATKEVICIUS, K. Investigation of Foreign Languages Models. *Elektronika Ir Elektrotechnika (Electronics and Electrical Engineering)*. 2012, vol. 18, no. 10, pp.53-56.
148. RUDŽIONIS, A., RATKEVIČIUS, K., RUDŽIONIS, V., Telekomunikacinės balso paslaugos. *Informacinės technologijos:* konferencijos pranešimų medžiaga. Kaunas, Technologija. 2003, p. 49 -54.
149. DUNN, M. *Pro Microsoft Speech Server 2007: Developing Speech Enabled Applications with .NET*. Apress, 2007, p. 275.
150. Universal Phone Set (UPS). [online]. 2014. [Retrieved 2017-05-15]. Access: http://msdn.microsoft.com/en-us/library/hh361647.aspx.
151. *ICD-10-CM Official Guidelines for Coding and Reporting* [online]. 2019. [Retrieved 2019-04-15]. Access: https://www.cms.gov/Medicare/Coding/ICD10/Downloads/2019-ICD10-Coding-Guidelines-.pdf .
152. SCHULTZ, T., WAIBEL, A. Language Independent and Language Adaptive Acoustic Modelling for Speech Recognition. *Speech Communication*. 2001, vol. 35, no. 1–2, pp. 31–51.
153. ZGANK, A. et al. The COST278 MASPER initiative – croslingual speech recognition with large telephone databases. *Proc. of the 4th International Conference on Language Resources and Evaluation (LREC'04)*. 2004, pp. 2107–2110.
154. KASPARAITIS, P. Lithuanian Speech Recognition Using the English recognizer. *Informatica*. 2008, vol. 19, no. 4, pp. 505–516.
155. NATO phonetic alphabet [online]. 2014. [Retrieved 2017-06-15]. Access: http://en.wikipedia.org/wiki/NATO_phonetic_alphabet.
156. LAURINČIUKAITĖ, S., TELKSNYS, L., KASPARAITIS, P., KLIUKIENĖ, R., PAUKŠTYTĖ, V. Lithuanian Speech Corpus Liepa for Development of Human-Computer Interfaces Working in Voice Recognition and Synthesis Mode. *INFORMATICA*. 2018, vol. 29, No. 3, pp. 487–498.
157. BARTIŠIŪTĖ, G., PAŠKAUSKAITĖ, G., RATKEVIČIUS, K. Investigation of disease codes recognition accuracy. *Proceedings of the 9th international conference on Electrical and Control Technologies, ECT 2014*. Kaunas University of Technology, Kaunas: Technologija. 2014, pp. 60-63.
158. LAURINČIUKAITĖ, S. *Acoustic modeling of Lithuanian speech recognition*. Ph.D. Thesis. Vilnius, 2008.
159. WANG, Y., WANG, H., GU, Z.-G. A Survey of Data Mining Software Used for Real Projects. *International Workshop on Open-Source Software for Scientific Computation (OSSC), Beijing*. 2011, pp. 94- 97.
160. HOFMANN, M. and KLINKENBERG, R. RapidMiner: Data Mining Use Cases and Business Analytics Applications. *Boca Raton: CRC Press*. 2013.
161. ZHAO, R. Reference Card for Data Mining [online]. 2016. [Retrieved 2017-06-15]. Access: http://www.rdatamining.com/docs/r-reference-card-for-data-mining.
162. BERTHOLD, M. R., CEBRON, N., DILL, F., GABRIEL, T. R, KÖTTER, T., MEINL, T. et al. KNIME: The Konstanz Information Miner. *Data Analysis, Machine Learning and Applications*. Springer, Berlin, Heidelberg. 2008, pp. 319–326.
163. HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P. and WITTEN I. H. The WEKA data mining software: an update. *SIGKDD Explorations.* 2009, vol. 11, no. 1, pp. 10–18.
164. DEMŠAR, J., CURK, T. and ERJAVEC A. Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research*. 2013, vol. 14, pp. 2349−2353.
165. JOVIC. A., BRKIC. K., BOGUNOVIC. N. An Overview of free software tools for general data mining. 37th International Convention on Information and Communication

Technology, *Electronics and Microelectronics (MIPRO*). Opatija, Croatia. 2014, pp.1112-1117.
166. CHEN, X., WILLIAMS, G., XU, X. A Survey of Open Source Data Mining Systems. *Emerging Technologies in Knowledge Discovery and Data Mining, Springer*. 2007, vol. 4819, pp.3-14.
167. WAHBEN, A. H., Al-RADAIDEH, Q.A., ALKABI, M.N., SHAWAKFA, E.M. A Comparison Study between Data Mining Tools over some Classification Methods. *International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence*. 2011, vol. 0(3), pp. 18–25.
168. PAULAUSKIENĖ, K., KURASOVA, O. Duomenų tyrybos sistemų galimybių tyrimas įvairių apimčių duomenims analizuoti. *Informacijos mokslai*. 2013, 65, pp.85-94.
169. JOVIC, A., BOGUNOVIC, N. Feature Set Extension for Heart Rate Variability Analysis by Using Non-linear, Statistical and Geometric Measures. Proceedings of the 31st International Conference on ITI. 2009, pp. 35-40.
170. SATHYADEVAN, S., NAIR, R.R. Comparative Analysis of Decision Tree Algorithms: ID3, C4.5 and Random Forest. In: Jain L., Behera H., Mandal J., Mohapatra D. (eds) *Computational Intelligence in Data Mining - Volume 1. Smart Innovation, Systems and Technologies*. Springer, New Delhi. 2015, vol. 31.
171. WU, X., KUMAR, V., QUINLAN, J.R. et al. Top 10 algorithms in data mining. *Knowledge and Information Systems.* Springer. 2007, vol. 14, issue 1, pp. 1-37.

9. CURRICULUM VITAE

Name: Gintarė
Surname: Žekienė
E-mail: Gintare.zekiene@ktu.lt;
Gintare.bartisiute@gmail.com

Education

Date	Degree	Establishment
2012–present	PhD, Informatics Engineering (07T)	Kaunas University of Technology, Faculty of Electrical and Electronics Engineering
2008–2011	Master of Electrical Engineering	Kaunas University of Technology, Faculty of Electrical and Electronics Engineering
2005–2008	Bachelor of Electrical Engineering	Kaunas University of Technology, Faculty of Electrical and Control Engineering,
1992–2005	High school	Kaunas Santara gymnasium

Areas of research

- Speech recognition, Language models;
- Energy market research.

Work experience

Date	Position	Establishment
2016–present	Lecturer	Kaunas University of Technology, faculty of informatics
2012–2016	Academic assistant	Kaunas University of Technology, Faculty of Electrical and Electronics Engineering
2014	Young researcher	Kaunas University of Technology, Information Systems Design Technology Center
2012–2013	Young researcher	Kaunas University of Technology, Speech Signal Research Science Laboratory
2011–2012	Electricity sales manager	UAB “ Vilnius energy”

10. LIST OF RESEARCH AND OTHER PUBLICATIONS

Articles published in journals belonging to international scientific databases

Indexed in the Web of Science with Impact Factor

1. **Bartišiūtė, Gintarė**; Ratkevičius, Kastytis. Speech server based Lithuanian voice commands recognition // Elektronika ir elektrotechnika = Electronics and electrical engineering. Kaunas: KTU. ISSN 1392-1215. 2012, Vol. 18, no. 10, p. 53-56. [Science Citation Index Expanded (Web of Science); INSPEC; Computers & Applied Sciences Complete; Central & Eastern European Academic Source] [Sc. fields: 01T]. [Contribution: 0,500]. [IF (E): 0,411 (2012)]
2. Rudžionis, Vytautas Evaldas; Raškinis, Gailius; Maskeliūnas, Rytis; Rudžionis, Algimantas Aleksandras; Ratkevičius, Kastytis; **Bartišiūtė, Gintarė**. Web services based hybrid recognizer of Lithuanian voice commands // Elektronika ir elektrotechnika = Electronics and electrical engineering. Kaunas: KTU. ISSN 1392-1215. 2014, Vol. 20, no. 9, p. 50-53. [Science Citation Index Expanded (Web of Science); Inspec; Computers & Applied Sciences Complete; Central & Eastern European Academic Source; Scopus] [Sc. fields: 01T]. [Contribution: 0,167]. [IF (E): 0,561 (2014)]
3. **Bartišiūtė, Gintarė**; Paškauskaitė, Gintarė; Ratkevičius, Kastytis. Advanced Recognition of Lithuanian Digit Names Using Hybrid Approach // Electronics and electrical engineering. Kaunas: KTU. ISSN 1392-1215. eISSN 2029- 5731. 2018, vol. 24, iss. 2, p. 70-73. DOI: 10.5755/j01.eie.24.2.20638. [Science Citation Index Expanded (Web of Science); Scopus] [IF: 0,684; AIF: 3,195; IF/AIF: 0,214; Q4 (2018, InCites JCR SCIE) [Sc. fields: 01T]. [Contribution: 0,333]

Publications in other international databases

1. **Bartišiūtė, Gintarė**; Ratkevičius, Kastytis. Investigation of Lithuanian digit names recignition accuracy // Electrical and control technologies: proceedings of the 8th international conference on electrical and control technologies ECT 2013, May 2-3, 2013, Kaunas, Lithuania / Kaunas University of Technology, IFAC Committee of National Lithuanian Organisation, Lithuanian Electricity Association. Kaunas: Technologija. ISSN 1822-5934. 2013, p. 9-12. [Conference Proceedings Citation Index] [Sc. fields: 01T]. [Contribution: 0,500]
2. Rudžionis, Vytautas; Raškinis, Gailius; Ratkevičius, Kastytis; Rudžionis, Algimantas Aleksandras; **Bartišiūtė, Gintarė**. Medical – pharmaceutical information system with recognition of Lithuanian voice commands // Human language technologies – the Baltic perspective: proceedings of the 6th international conference, Baltic HLT 2014, Kaunas, Lithuania, September 26-27, 2014 / edited by A. Utka, G. Grigonytė, J. Kapočiūtė-Dzikienė, J.

Vaičenonienė. Amsterdam: IOS Press. (Frontiers in artificial intelligence and applications, vol. 268, ISSN 0922-6389), ISBN 9781614994411. p. 40-45. [Conference Proceedings Citation Index – Science (Web of Science)] [Sc. fields: 07T]. [Contribution: 0,200]

3. **Bartišiūtė, Gintarė**; Paškauskaitė, Gintarė; Ratkevičius, Kastytis. Investigation of disease codes recognition accuracy // Proceedings of the 9th international conference on Electrical and Control Technologies, ECT 2014 / Kaunas University of Technology, IFAC Committee of National Lithuanian Organisation, Lithuanian Electricity Association. Kaunas: Technologija. ISSN 1822-5934. 2014, p. 60-63. [Scopus] [Sc. fields: 01T]. [Contribution: 0,333]
4. **Bartišiūtė, Gintarė**; Ratkevičius, Kastytis; Paškauskaitė, Gintarė. Hybrid recognition technology for isolated voice commands // Information systems architecture and technology: Proceedings of 36th international conference on information systems architecture and technology, ISAT 2015, Part 4 / Zofia Wilimowska, Leszek Borzemski, Adam Grzech, Jerzy Świątek, eds. Cham: Springer, 2016. (Advances in intelligent systems and computing, vol. 432, ISSN 2194-5357), ISBN 9783319285658. p. 207-216. [Conference Proceedings Citation Index-Science; SpringerLINK] [Sc. fields: 07T]. [Contribution: 0,333]
5. Ratkevičius, Kastytis; Paškauskaitė, Gintarė; **Bartišiūtė, Gintarė**. Recognition of ICD-10 codes by combining two recognizers // Frontiers in artificial intelligence and applications: Human language technologies – the Baltic perspective: proceedings of the seventh international conference Baltic HLT 2016 / edited by Inguna Skadiņa, Roberts Rozis. Amsterdam: IOS Press. ISSN 0922-6389. 2016, vol. 289, p. 51-58. [Scopus] [Sc. fields: 01T]. [Contribution: 0,333]
6. **Bartišiūtė, Gintarė**; Paškauskaitė, Gintarė. Šnekos atpažintuvų sujungimo galimybių tyrimas // E2TA-2015: Elektronika, elektra, telekomunikacijos, automatika: 12-osios studentų mokslinės konferencijos pranešimų medžiaga = 12th student scientific conference on electronics, energy, telecommunications and automation. Kaunas: Kauno technologijos universitetas, 2015, ISBN 9786090211335. p. 20-23. [Sc. fields: 07T]. [Contribution: 0,500]

11. ACKNOWLEDGEMENTS

Undertaking this PhD has been a life-changing experience, and it would not have been possible without the overwhelming amount of assistance and guidance that I have received from all of those who have supported me. Their constant encouragement has enabled me to navigate the most turbulent period in my life so far.

First and foremost, I am extremely grateful to my supervisor – associate professor Kastytis Ratkevičius – for his invaluable advice, continuous guidance, and no small amount of patience throughout my PhD. His constant encouragement, both academic and otherwise, has been an indispensable source of comfort. I feel privileged to have met and learnt from him, not only for his professional knowledge but also for his moral values.

I would also like to thank all of the co-authors of the articles that I have published for collaborating with me and sharing their knowledge and expertise. I greatly appreciate the reviewers of this thesis and thank them for their detailed comments and invaluable advice.

I am extremely thankful for the encouragement and support that I have received from my friends and family throughout my studies, particularly my mother, Ramute, and my husband, Julius. It is difficult to imagine having finished this thesis without their guidance. Finally, I'd like to thank my son, Kasparas, whose companionship enabled the writing of this thesis. Although he only appeared half way through, his presence was felt throughout, and as such it feels only fair to name him as a co-author. I hope he will be proud.

Sincerely

Gintarė Žekienė

Annex 1

Table 1. English language digit transcriptions

Digit	Transcription
0	Nulis; nuhlihs**;** nuhliys**;** nuwlihs; nuwliys.
1	Vienas; viyaxnahs; viyaxnaxs; viyaxnaas; viyahnahs; viyahnaxs; Viyahnaas; viyehnahs; viyehnaxs; viyehnaas.
2	Du; Duuh; duuw; duh**;** duw.
3	Trys; tris; trees; triys**;** trihs.
4	Keturi; Kehtuhraih; ketthurriy; kehtuhrih; kehtuhriy; kehtuwrih; Kehtuwriy; kaetuhrih; kaetuhriy; kaetuwrih; kaetuwriy.
5	Penki; pehnkih; pehngkih**;** pehngkiy**;** paengkih**;** paengkiy.
6	Sheshi; shehshih**;** shehshiy**;** shaeshih**;** shaeshiy.
7	Septyni; sehptinih; sehptiynih**;** sehptiyniy**;** sehptihnih**;** sehptihniy**;** saeptiynih**;** saeptiyniy**;** saeptihnih**;** saeptihniy.
8	Ashtuoni; ahshtuhaanih; ahshtuwaxnih; ahshtuwaxniy; ahshtuwahnih; ahshtuwahniy; axshtuwaxnih; axshtuwaxniy; Axshtuwahnih; axshtuwahniy; aashtuwaxnih; aashtuwaxniy; aashtuwahnih; aashtuwahniy.
9	Devyni; dehvinih; dehviynih; dehviyniy; dehvihnih; dehvihniy; Daeviynih; daeviyniy; daevihnih; daevihniy.

Table 2. German language digit transcriptions

Digit	Transcription
0	Nuhlihs; Nulis; Nulihs; Nuhlis; Nuhlys; Nulys.
1	Vihehnas; Vienas; Vihaxnas; Viyenas; Vihyenas; Vihyehnas; Vyenas; Vyehnas.
2	Du; Duh; Duuh; Duw.
3	Trhies; Trys; Tries.
4	Kaxtuhrhih; Kehtuhrhih; Keturi; Keturhih; Keturih; Kehtuhrih.
5	Paxnkih; Pehnkih; Penki; Penkih; Pehnki.
6	Sheshi; Shehshih; Shaxshih; Shehshi; Sheshih; Scheschi; Schehschi; Schehschih; Scheschih.
7	Sseptyni; Saxptienih; Sehptienih; Septyni; Ssehptienih; Ssehptynih; Sseptieni; Sseptienih.
8	Ashtuhaonih; Ashtuoni; Ashtuhonih; Ashtuhoni; Ashtuonih.
9	Dehvienih; Devyni; Daxvienih; Devienih; Dehvieni; Dehvyni; Dehvynih; Devieni; Devini; Devinih; Devynih.

Table 3. French language digit transcriptions

Digit	Transcription
0	Nulis; Nouluece; Noulice; Noulihce; Noulihss; Nouliss.s.
1	Vienas; Viehnass; Viehnaass; Viehnace; Vyehnace; Vyehnass..
2	Du; Dou; Duh; Douh; Dous.
3	Trys; Tryss; Tryce; Tryhss; Tryhce; Triss; Trihss; Trihce; Trice
4	Keturi; Kehturi; Kehtouri; Kehtourhi; Kehtourih; Kehtouryh; Kehtourhy;Kehtoury; Ketoury; Ketouri; Getoudi.
5	Penki; Pinki; Pehnki; Pinkih; Pinky; Pinkyh
6	Sheshi; Shechi; Chechi; Chehchi; Chehchih; Chechy; Chehchy; Chechyh..
7	Septyni; Sehptyni; Sehptueni; Sehptini; Sehptyhni; Sehptyhnih; Sehptihnih; Sehptyhnyh; Septini.
8	Ashtuoni; Achtuoni; Achtuonih; Achtuony; Achtuonyh; Achtuhohni; Achtouohni.
9	Devyni; Devynih; Dehvyni; Dehvini; Dehvinih

Annex 2

Table 1. Full grammar for selection of names and words corresponding to 26 Latin letters for identification (NAMES1)

Letter	Name
A	Adomas, Agota, Aleksas, Andrius, Antanas, Arnoldas, Artūras, Asta, Aurelija, Austeja
B	Barbora, Bernardas, Beata, Benas, Benediktas, Birutė, Boleslovas, Božena, Brigita, Bronius
C	Cecilija, Celestas, Cezaris
D	Daiva, Dalia, Danguolė, Danutė, Daumantas, Deimantė, Dominykas, Donatas, Dovilė, Dovydas
E	Edgaras, Egidija, Eimantas, Elena, Eligijus, Elvyra, Emilija, Erika, Evaldas, Evelina
F	Fausta, Felicija, Feliksas, Filomena, Florijona, Fortūna, Fridrikas
G	Gabija, Gabrielė, Gediminas, Gerda, Gintaras, Gintautas, Gitana, Goda, Gražvydas, Greta
H	Hamletas, Hansas, Haroldas, Henrikas, Heraklis, Hermanas, Horacijus
I	Ignas, Ilona, Indraja, Indrė, Inesa, Irena, Irmantas, Iveta, Izabelė, Izaokas
J	Jadvyga, Joana, Jokūbas, Jolanta, Jonas, Jovita, Julija, Juozapas, Jūratė, Justas
K	Kajus, Kamilė, Karolis, Kazys, Kęstutis, Klaudijus, Kornelija, Kostas, Kotryna, Kristina
L	Laima, Laurynas, Leonas, Lilija, Linas, Liucija, Liutauras, Liveta, Loreta, Lukas
M	Mantas, Margarita, Marius, Marytė, Martynas, Matas, Mažvydas, Mindaugas, Mykolas, Modestas
N	Natalija, Nedas, Neimantas, Nerijus, Neringa, Nijolė, Nikolajus, Nojus, Nomeda, Normantas
O	Odeta, Odrė, Ofelija, Oksana, Olegas, Ona, Orinta, Oskaras, Otilija, Ovidijus
P	Palmira, Patricija, Patrikas, Paulius, Petras, Pijus, Pilypas, Povilas, Pranas, Prudencijus
Q	Kju, Kventinas
R	Radvilė, Raminta, Ramūnas, Renata, Ričardas, Rima, Rytis, Rokas, Rolandas, Rūta
S	Sandra, Saulė, Sigitas, Simona, Skirmantas, Sofija, Solveiga, Sonata, Steponas, Svajūnas
T	Tadas, Tatjana, Tauras, Tautvilė, Tautvydas, Teodoras, Teresė, Timas, Titas, Tomas
U	Ubalda, Ugnė, Uldis, Ulijona, Unė, Uosis, Urbonas, Ursinas, Urðulė, Urtė
V	Vacys, Vaida, Valerija, Veronika, Viktoras, Vilius, Viltė, Vincas, Vygantas, Vytautas
W	Dabalju, Wašington
X	Iksas
Y	Ygrek, Ygrik, Yla, Ygrekas
Z	Zacharijus, Zenonas, Zigfridas, Zigmas, Zilbertas, Zinaida ,Zita, Zofija, Zoja, Zosė

Annex 3

Table 1. List of speakers used in digit names recognition cross-validation

1 FOLD		2 FOLD		3 FOLD		4 FOLD		5 FOLD	
Training	**Testing**	**Training**	**Testing**	Training	Testing	Training	Testing	Training	Testing
FAGNGRA	FIEVVIS	FIEVVIS	FAGNGRA	FIEVVIS	FGINGED	FIEVVIS	FRUTNAN	FIEVVIS	MMODSLE
FAGNVIN	FJUSKIN	FJUSKIN	FAGNVIN	FJUSKIN	FIEVJUR	FJUSKIN	FSIMMEI	FJUSKIN	MRIMAPA
FAISIZI	FUGNBUC	FUGNBUC	FAISIZI	FUGNBUC	FIEVSAB	FUGNBUC	FVAIVAI	FUGNBUC	MVYGVAI
FAISZYM	FUGNNOV	FUGNNOV	FAISZYM	FUGNNOV	FKAMMOS	FUGNNOV	FVANPEC	FUGNNOV	FVIONAB
FAUSNEM	FZIVSTA	FZIVSTA	FAUSNEM	FZIVSTA	FLAUZET	FZIVSTA	FVILVAI	FZIVSTA	FRAISAV
FDAILOI	MLINJUR	MLINJUR	MDAIGUS	MLINJUR	MEDGVOL	MLINJUR	MKAZANU	MLINJUR	FDAILOI
FGINGED		FGINGED		FAGNGRA		FAGNGRA		FAGNGRA	
FIEVJUR		FIEVJUR		FAGNVIN		FAGNVIN		FAGNVIN	
FIEVSAB		FIEVSAB		FAISIZI		FAISIZI		FAISIZI	
FKAMMOS		FKAMMOS		FAISZYM		FAISZYM		FAISZYM	
FLAUZET		FLAUZET		FAUSNEM		FAUSNEM		FAUSNEM	
FRAISAV		FRAISAV		FDAILOI		FDAILOI		MDAIGUS	
FRUTNAN		FRUTNAN		FRUTNAN		FGINGED		FGINGED	
FSIMMEI		FSIMMEI		FSIMMEI		FIEVJUR		FIEVJUR	
FVAIVAI		FVAIVAI		FVAIVAI		FIEVSAB		FIEVSAB	
FVANPEC		FVANPEC		FVANPEC		FKAMMOS		FKAMMOS	
FVILVAI		FVILVAI		FVILVAI		FLAUZET		FLAUZET	
FVIONAB		FVIONAB		FVIONAB		FRAISAV		MEDGVOL	
MDAIGUS		FDAILOI		MDAIGUS		MDAIGUS		FRUTNAN	
MEDGVOL		MEDGVOL		FRAISAV		MEDGVOL		FSIMMEI	
MKAZANU		MKAZANU		MKAZANU		FVIONAB		FVAIVAI	
MMODSLE		MMODSLE		MMODSLE		MMODSLE		FVANPEC	
MRIMAPA		MRIMAPA		MRIMAPA		MRIMAPA		FVILVAI	
MVYGVAI		MVYGVAI		MVYGVAI		MVYGVAI		MKAZANU	

Annex 4

Table 1A. Phoneme set (v ie n a s d u t r y k e i p sh uo l sil sp) RA

Digit1						
Command	**Phoneme set distribution**	**RA, %**				
		1-fold	**2-fold**	**3-fold**	**4-fold**	**5-fold**
VIENAS	v ie n a s sp	73.3	88.3	85.0	100	57,5
DU	d u sp	0.0	0.0	3.3	0.0	0,0
TRYS	t r y s sp	33.3	65.8	50.8	43.3	35,0
KETURI	k e t u r i sp	25.0	17.5	34.2	15.0	27,5
PENKI	p e n k i sp	12.5	5.8	23.3	3.3	6,7
SHESHI	sh e sh i sp	85.8	86.7	100	97.5	97,5
SEPTYNI	s e p t y n i sp	98.3	100	92.5	90.8	100
ASHTUONI	a sh t uo n i sp	100	97.5	96.7	100	90,0
DEVYNI	d e v y n i sp	92.5	80.8	94.2	99.2	70,0
NULIS	n u l i s sp	100	99.2	99.2	94.2	93,3
Average RA, %		**62.08**	**64.17**	**67.92**	**64.33**	**56.75**
Overall 5 folds average RA, %		**63.05**				

Table 2A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk sp) – SAMPA

Digit2						
Command	**Phoneme set distribution**	**RA, %**				
		1-fold	**2-fold**	**3-fold**	**4-fold**	**5-fold**
VIENAS	vm ie n a s sp	100	100	100	100	97,5
DU	d uk sp	0.8	0.0	1.7	0.0	0,0
TRYS	tm rm y s sp	95.0	95.0	95.0	95.0	94,2
KETURI	km e t u rm ik sp	99.2	84.2	79.2	71.7	72,5
PENKI	pm e nk km ik sp	97.5	95.8	100	72.5	93,3
SHESHI	shm e shm ik sp	84.4	75.0	96.7	85.8	71,7
SEPTYNI	sm e pm tm ii nm ik sp	95.0	100	100	91.7	99,2
ASHTUONI	a sh t uo nm ik sp	100	100	100	100	100
DEVYNI	dm e vm ii nm ik sp	100	90.8	100	100	90,8
NULIS	n uk lm i s sp	98.3	97.5	96.7	95.0	97,5
AVERAGE RA, %		**87.0**	**83.83**	**86.92**	**81.17**	**81.67**
Overall 5 folds average RA, %		**84.12**				

Table 3A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk **ud** sp). Includes a new allophone *ud,* in command DU

Digit3		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	70.8

TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	99.2
PENKI	pm e nk km ik sp	97.5
SHESHI	shm e shm ik sp	84.2
SEPTYNI	sm e pm tm ii nm ik sp	95.8
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	100
NULIS	n uk lm i s sp	98.3
AVERAGE RA, %		**94.08**

Table 4A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud **ish** sp). Includes a new allophone *ish*, in command SESI

Digit4		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	90.0
DU	d ud sp	99.2
TRYS	tm rm y s sp	95
KETURI	km e t u rm ik sp	97.5
PENKI	pm e nk km ik sp	90.0
SHESHI	shm e shm ish sp	88.3
SEPTYNI	sm e pm tm ii nm ik sp	89.2
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	95.8
AVERAGE RA, %		**94.42**

Table 5A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish **esh** sp). Includes a new allophone *esh*, in command SESI

Digit5		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	64.2
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	97.5
PENKI	pm e nk km ik sp	98.3
SHESHI	shm esh shm ish sp	96.7
SEPTYNI	sm e pm tm ii nm ik sp	95.8
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	98.3
AVERAGE RA, %		**94.5**

Table 6A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh sp). In command SEPTYNI, phoneme *ii* is changed into phoneme *y*

Digit6		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	66.7
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	98.3
PENKI	pm e nk km ik sp	95.0
SHESHI	shm esh shm ish sp	96.7
SEPTYNI	sm e pm tm y nm ik sp	95.0
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	100
AVERAGE RA, %		**95.58**

Table 7A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh **ek** sp). Includes a new phoneme *ek*, in command PENKI

Digit7			
Command	**Phoneme set distribution**	**RA, %**	
		1-fold	**5-fold**
VIENAS	vm ie n a s sp	100	97.2
DU	d ud sp	69.2	43.3
TRYS	tm rm y s sp	95.0	90.0
KETURI	km e t u rm ik sp	96.7	57.5
PENKI	pm ek nk km ik sp	98.3	80.8
SHESHI	shm esh shm ish sp	95.8	100
SEPTYNI	sm e pm tm y nm ik sp	95.0	99.2
ASHTUONI	a sh t uo nm ik sp	100	100
DEVYNI	dm e vm ii nm ik sp	99.2	80.8
NULIS	n uk lm i s sp	99.2	92.5
AVERAGE RA, %		**94.92**	**85.17**

Table 8A. Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek sp). Diphthong *ie* in command VIENAS is separated into two phonemes *ik* and *e*

Digit8		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ik e n a s sp	84.2
DU	d ud sp	80.8
TRYS	tm rm y s sp	94.2
KETURI	km e t u rm ik sp	95
PENKI	pm ek nk km ik sp	98.3
SHESHI	shm esh shm ish sp	95.8
SEPTYNI	sm e pm tm y nm ik sp	95.0
ASHTUONI	a sh t uo nm ik sp	100

DEVYNI	dm e vm ii nm ik sp	100
NULIS	n uk lm i s sp	100
AVERAGE RA, %		**93.92**

Table 9A. Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek sp) Diphthong *ie* in command VIENAS is separated into two phonemes *i* and *ek*

Digit9						
Command	**Phoneme set distribution**	**RA, %**				
		1-fold	**2-fold**	**3-fold**	**4-fold**	**5-fold**
VIENAS	vm i ek n a s sp	100	100	100	100	98,3
DU	d ud sp	79.2	60.0	92.5	98.3	82,5
TRYS	tm rm y s sp	95	91.7	95.0	95.0	90,8
KETURI	km e t u rm ik sp	95.8	71.7	60.0	63.3	60,0
PENKI	pm ek nk km ik sp	96.7	74.2	92.5	79.2	58,3
SHESHI	shm esh shm ish sp	95.8	96.7	100	100	100
SEPTYNI	sm e pm tm y nm ik sp	96.7	100	100	92.5	99,2
ASHTUONI	a sh t uo nm ik sp	100	100	100	100	100
DEVYNI	dm e vm ii nm ik sp	99.2	97.5	99.2	100	92,5
NULIS	n uk lm i s sp	100	97.5	96.7	94.2	95,0
AVERAGE RA, %		**95.83**	**88.92**	**93.58**	**92.25**	**87.67**
Overall 5 folds Average RA, %		**91.65**				

Table 10A. Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek sp). Includes phoneme *e*, in command PENKI

Digit10		
Command	**Phoneme set distribution**	**RA, %**
		5-fold
VIENAS	vm i ek n a s sp	98.3
DU	d ud sp	82.5
TRYS	tm rm y s sp	91.7
KETURI	km e t u rm ik sp	60.8
PENKI	pm e nk km ik sp	67.5
SHESHI	shm esh shm ish sp	100
SEPTYNI	sm e pm tm y nm ik sp	99.2
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	94.2
NULIS	n uk lm i s sp	98.3
AVERAGE RA, %		**89.25**

Table 11A. Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek en sp). Includes a new phoneme *en*, in command PENKI

Digit11		
Command	**Phoneme set distribution**	**RA, %**
		5-fold
VIENAS	vm i ek n a s sp	99.2
DU	d ud sp	82.5
TRYS	tm rm y s sp	90.8
KETURI	km e t u rm ik sp	56.7
PENKI	pm en nk km ik sp	80.8
SHESHI	shm esh shm ish sp	100
SEPTYNI	sm e pm tm y nm ik sp	99.2
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	91.7
NULIS	n uk lm i s sp	98.3
AVERAGE RA, %		**89.92**

Table 12A. Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek en et sp). Includes a new phoneme *et*, in command KETURI

Digit12		
Command	**Phoneme set distribution**	**RA, %**
		5-fold
VIENAS	vm i ek n a s sp	99.2
DU	d ud sp	84.2
TRYS	tm rm y s sp	91.7
KETURI	km et t u rm ik sp	66.7
PENKI	pm en nk km ik sp	87.5
SHESHI	shm esh shm ish sp	100
SEPTYNI	sm e pm tm y nm ik sp	99.2
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	96.7
NULIS	n uk lm i s sp	100
AVERAGE RA, %		**92.50**

Table 13A. Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek en et ir sp). Includes a new phoneme *ir*, in command KETURI

Digit13			
Command	**Phoneme set distribution**	**RA, %**	
		2-fold	**5-fold**
VIENAS	vm i ek n a s sp	99.2	99.2
DU	d ud sp	65.0	84.2
TRYS	tm rm y s sp	92.5	93.3

KETURI	km et t u rm ir sp	85.0	70.8
PENKI	pm en nk km ik sp	91.7	88.3
SHESHI	shm esh shm ish sp	97.5	100
SEPTYNI	sm e pm tm y nm ik sp	100	99.2
ASHTUONI	a sh t uo nm ik sp	100	100
DEVYNI	dm e vm ii nm ik sp	99.2	98.3
NULIS	n uk lm i s sp	98.3	100
AVERAGE RA, %		**92.83**	**93.33**

Table 14A. Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek en et ir sp). Additional phoneme *ud* is added to command DU

Digit14			
Command	**Phoneme set distribution**	**RA, %**	
		2-fold	**5-fold**
VIENAS	vm i ek n a s sp	99.2	99.2
DU	d ud ud sp	90.8	94.2
TRYS	tm rm y s sp	92.5	93.3
KETURI	km et t u rm ir sp	85.0	71.7
PENKI	pm en nk km ik sp	92.5	88.3
SHESHI	shm esh shm ish sp	97.5	100
SEPTYNI	sm e pm tm y nm ik sp	100	98.3
ASHTUONI	a sh t uo nm ik sp	100	100
DEVYNI	dm e vm ii nm ik sp	99.2	98.3
NULIS	n uk lm i s sp	98.3	99.2
AVERAGE RA, %		**95.50**	**94.25**

Table 15A. Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek en et ir ke sp). Includes a new phoneme *ke*, in command KETURI

Digit15		
Command	**Phoneme set distribution**	**RA, %**
		5-fold
VIENAS	vm i ek n a s sp	99.2
DU	d ud ud sp	93.3
TRYS	tm rm y s sp	93.3
KETURI	ke et t u rm ir sp	57.5
PENKI	pm en nk km ik sp	91.7
SHESHI	shm esh shm ish sp	100
SEPTYNI	sm e pm tm y nm ik sp	99.2
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	98.3
NULIS	n uk lm i s sp	99.2
AVERAGE RA, %		**93.17**

Table 16A. Phoneme set (vm n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek en et ir ri sp). Includes a new phoneme *ri*, in command KETURI

Digit16						
Command	**Phoneme set distribution**	**RA, %**				
		1-fold	**2-fold**	**3-fold**	**4-fold**	**5-fold**
VIENAS	vm i ek n a s sp	100	99.2	100	100	99,2
DU	d ud ud sp	100	90.8	95.8	100	94,2
TRYS	tm rm y s sp	95.0	94.2	95.0	95.0	92,5
KETURI	km et t u ri ir sp	100	91.7	93.3	82.5	86,7
PENKI	pm en nk km ik sp	100	94.2	100	96.7	90,0
SHESHI	shm esh shm ish sp	96.7	97.5	100	100	100
SEPTYNI	sm e pm tm y nm ik sp	96.7	100	98.3	93.3	98,3
ASHTUONI	a sh t uo nm ik sp	100	100	100	100	100
DEVYNI	dm e vm ii nm ik sp	100	98.3	99.2	100	98,3
NULIS	n uk lm i s sp	100	99.2	98.3	96.7	98,3
AVERAGE RA, %		**98.84**	**96.51**	**97.99**	**96.42**	**95.75**
Overall 5 folds Average RA, %		**97.1**				

Table 17A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek **o** sp). A new phoneme *o* is added to the set. Diphthong *uo* in command ASTUONI is separated into two phonemes *u* and *o*

Digit17		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	63.3
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	93.3
PENKI	pm ek nk km ik sp	98.3
SHESHI	shm esh shm ish sp	96.7
SEPTYNI	sm e pm tm y nm ik sp	95.8
ASHTUONI	a sh t u o nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	100
AVERAGE RA, %		**94.17**

Table 18A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek o sp). In command ASTUONI, phoneme *u* is changed into *uk*

Digit18		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	68.3
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	96.7
PENKI	pm ek nk km ik sp	98.3
SHESHI	shm esh shm ish sp	97.5
SEPTYNI	sm e pm tm y nm ik sp	95.8
ASHTUONI	a sh t uk o nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	100
AVERAGE RA, %		**95.08**

Table 19A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh sp). In command SEPTYNI, phoneme *ii* is changed into *i*

Digit19		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	64.2
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	100
PENKI	pm e nk km ik sp	95.8
SHESHI	shm esh shm ish sp	95.8
SEPTYNI	sm e pm tm i nm ik sp	95.0
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	97.5
AVERAGE RA, %		**95.33**

Table 20A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh **yk** sp)). A new phoneme *yk* is added to the set. Phoneme *ii* in command SEPTYNI is changed into *yk*. Also previous phoneme *ik* is changed into *i*

Digit20		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	62.5
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	98.3

PENKI	pm e nk km ik sp	99.2
SHESHI	shm esh shm ish sp	96.7
SEPTYNI	sm e pm tm yk nm i sp	97.5
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	98.3
AVERAGE RA, %		**94.67**

Table 21A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek **yk** sp). A new phoneme *yk* is added to the set. Phoneme *ii* in command DEVYNI is changed into *yk*. Also previous phoneme *ik* is changed into *i*

Digit21		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	69.2
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	96.7
PENKI	pm ek nk km ik sp	89.3
SHESHI	shm esh shm ish sp	95.8
SEPTYNI	sm e pm tm y nm ik sp	98.3
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm yk nm i sp	100
NULIS	n uk lm i s sp	98.3
AVERAGE RA, %		**95.17**

Table 22A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek yk sp). Command DEVYNI includes two types of phoneme sets

Digit22		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	69.2
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	96.7
PENKI	pm ek nk km ik sp	98.3
SHESHI	shm esh shm ish sp	95.8
SEPTYNI	sm e pm tm y nm ik sp	95.0
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm yk nm i sp dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	100
AVERAGE RA, %		**94.92**

Table 23A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek o sp). Command ASTUONI includes two types of phoneme sets

Digit23		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	63.3
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	93.3
PENKI	pm ek nk km ik sp	98.3
SHESHI	shm esh shm ish sp	96.7
SEPTYNI	sm e pm tm y nm ik sp	95.8
ASHTUONI	a sh t u o nm ik sp a sh t uk o nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	100
AVERAGE RA, %		**94.17**

Table 24A. Phoneme set (vm ie n a s d u tm rm y km e i pm shm uo lm sil t ii nk sh nm dm sm ik uk ud ish esh ek yk sp. Command SEPTYNI includes two types of phoneme sets

Digit24		
Command	**Phoneme set distribution**	**RA, %**
		1-fold
VIENAS	vm ie n a s sp	100
DU	d ud sp	69.2
TRYS	tm rm y s sp	95.0
KETURI	km e t u rm ik sp	96.7
PENKI	pm ek nk km ik sp	98.3
SHESHI	shm esh shm ish sp	95.8
SEPTYNI	sm e pm tm y nm ik sp sm e pm tm yk nm i sp	95.0
ASHTUONI	a sh t uo nm ik sp	100
DEVYNI	dm e vm ii nm ik sp	99.2
NULIS	n uk lm i s sp	100
AVERAGE RA, %		**94.92**

Annex 5

Table 1. List of speakers used for names and words recognition

Speaker	Number
FAGNRUM	0F
FEGLZAJ	1F
FIVEVAL	2F
FJULBAL	3F
MJURBIZ	4M
MLAUBAR	5M
MROKKUO	6M
MSARNEM	7M
FGINBAR	8F
MDARJEG	9M
FGINPAS	10F
FGINTRA	11F
FGRETUB	12F
FINDBEN	13F
FLAUKLU	14F
FMILRAU	15F
MANDMAR	16M
MJUOCES	17M
MKASRAT	18M
MZYGSVE	19M
FSEVBUT	20F

Table 2. Names and words 7-times cross-validation speaker distribution

1-FOLD		**2-FOLD**		**3-FOLD**		**4-FOLD**		**5-FOLD**	
Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
0F	8F	8F	0F	8F	3F	8F	6M	8F	11F
1F	9M	9M	1F	9M	4M	9M	7M	9M	12F
2F	16F	16F	2F	16F	5M	16F	10F	16F	13F
3F		3F		0F		0F		0F	
4M		4M		1F		1F		1F	
5M		5M		2F		2F		2F	
6M		6M		6M		3F		3F	
7M		7M		7M		4M		4M	
10F		10F		10F		5M		5M	
11F		11F		11F		11F		6M	
12F		12F		12F		12F		7M	
13F		13F		13F		13F		10F	
14F		14F		14F		14F		14F	
15F		15F		15F		15F		15F	
17M		17M		17M		17M		17M	
18M		18M		18M		18M		18M	
19M		19M		19M		19M		19M	
20M		20M		20M		20M		20M	

6-FOLD		7-FOLD	
Training	Testing	Training	Testing
0F	14F	8F	18M
1F	15F	9M	19M
2F	17M	16F	20M
3F		3F	
4M		4M	
5M		5M	
6M		6M	
7M		7M	
10F		10F	
11F		11F	
12F		12F	
13F		13F	
8F		14F	
9M		15F	
16F		17M	
18M		0F	
19M		1F	
20M		2F	

Table 3. Average names and words RA by varying number of states

Command	+0 states	+2 states	+3 states	+4 states	+6 states
Austėja	100	100	97.5	100	100
Boleslovas	100	100	100	98.3	98.3
Cecilija	100	100	100	100	100
Donatas	66.7	100	97.5	48.3	40
Eimantas	98.3	100	100	95	100
Fausta	26.7	98.3	97.5	100	100
Gražvydas	76.7	50	100	41.7	98.3
Hansas	43.3	100	47.5	100	100
Izaokas	100	100	100	100	8.3
Jonas	11.7	83.3	100	83.3	100
Karolis	58.3	100	100	100	11.7
Laima	8.3	75	100	50	100
Martynas	85	100	100	100	100
Nojus	10	75	100	100	100
Oskaras	71.7	100	100	100	16.7
Patrikas	90	91.7	100	50	96.7
Kju	0	1.7	55	95	95
Ričardas	31.7	91.7	2.5	31.7	80
Sandra	6.7	100	80	100	100
Teodoras	96.7	100	100	98.3	100
Ulijona	100	100	100	98.3	70
Vacys	18.3	70	97.5	100	100
Wašington	68.3	70	77.5	70	100
Xsas	0	0	65	60	90
Ygrekas	96.7	100	97.5	31.7	26.7
Zacharijus	100	100	95	100	100
RA, %	**60.20±14.85**	**84.87±10.79**	**88.85±8.85**	**82.75±9.97**	**81.99±12.19**

Table 4. Average names and words RA by varying number of Gaussian mixtures in states

Command	Additional states. + 2					Additional states. + 3				
	2 Mixtures	3 Mixtures	4 Mixtures	6 Mixtures	10 Mixtures	1 Mixture	2 Mixtures	3 Mixtures	4 Mixtures	6 Mixtures
Austėja	100	100	100	100	100	100	100	100	100	100
Boleslovas	100	100	100	100	100	100	100	100	100	98.3
Cecilija	100	100	100	100	100	100	100	100	100	100
Donatas	100	100	100	100	100	100	100	100	100	100
Eimantas	100	100	100	100	100	100	100	50	100	100
Fausta	100	100	100	96.7	100	100	100	100	100	100
Gražvydas	100	100	100	100	100	100	100	100	100	98.3
Hansas	100	100	100	100	100	100	100	95	100	100
Izaokas	98.3	98.3	100	100	100	100	100	100	100	100
Jonas	98.3	100	51.7	38.3	100	93.3	100	100	100	58.3
Karolis	96.7	75	100	100	100	100	100	100	100	100
Laima	96.7	91.7	90	81.7	100	100	98.3	98.3	100	70
Martynas	100	100	100	100	100	100	100	100	100	100
Nojus	98.3	100	73.3	96.7	100	98.3	100	100	100	100
Oskaras	85	66.7	95	100	100	100	100	100	100	95
Patrikas	100	100	100	100	100	95	100	96.7	100	100
Kju	83.3	91.7	88.3	88.3	86.7	71.7	86.7	95	85	95
Ričardas	100	100	100	100	100	96.7	98.3	88.3	100	100
Sandra	100	100	100	100	100	100	100	100	100	100
Teodoras	100	100	100	100	100	100	100	40	100	100
Ulijona	100	100	98.3	100	100	100	100	100	100	100
Vacys	85	91.7	98.3	1.7	91.7	68.3	98.3	78.3	98.3	93.3
Wašington	100	80	100	100	100	100	100	95	100	88.3
Xsas	70	8.3	58.3	38.3	100	60	96.7	81.7	36.7	96.7
Ygrekas	100	91.7	100	100	100	100	100	100	100	98.3
Zacharijus	100	100	100	100	100	100	100	98.3	100	100
RA, %	**96.6 ± 2.83**	**92.12 ± 7.38**	**94.35 ± 5.00**	**90.07 ± 9.48**	**99.17 ± 0.83**	**95.51 ± 4.19**	**99.2**	**93**	**96.9**	**95.8**

Command	Additional states. +4					
	2 Mixtures	3 Mixtures	4 Mixtures	6 Mixtures	10 Mixtures	16 Mixtures
Austėja	100	100	100	100	100	100
Boleslovas	100	100	100	98.3	100	100
Cecilija	100	100	100	100	100	100
Donatas	100	100	100	95	100	100
Eimantas	100	100	100	98.3	100	100
Fausta	100	100	100	100	100	100
Gražvydas	100	100	100	100	100	100
Hansas	100	100	100	100	100	100
Izaokas	100	100	100	96.7	100	100
Jonas	91.7	41.7	86.7	100	98.3	100
Karolis	100	100	100	100	100	100
Laima	100	100	100	100	100	100
Martynas	100	68.3	100	100	100	100
Nojus	100	100	100	100	100	100
Oskaras	100	56.7	100	61.7	100	100
Patrikas	100	78.3	100	86.7	100	100
Kju	85	95	86.7	95	95	95
Ričardas	100	75	100	78.3	100	100
Sandra	100	100	100	100	100	100
Teodoras	100	100	100	100	100	100
Ulijona	100	100	100	100	100	100
Vacys	66.7	100	98.3	100	100	98.3
Wašington	100	90	100	98.3	100	100
Xsas	91.7	98.3	95	85	100	100
Ygrekas	100	96.7	100	95	100	100
Zacharijus	100	100	100	98.3	100	100
RA, %	**97.5**	**92.3**	**98.7**	**95.6**	**99.7**	**99.7**

Annex 6

1-FOLD		2-FOLD		3-FOLD		4-FOLD		5-FOLD	
Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
D03	D48	D48	D03	D03	D27	D03	D44	D03	D58
D05	D54	D54	D05	D05	D28	D05	D45	D05	D61
D07	D87	D87	D07	D07	D29	D07	D78	D07	D62
D10	D88	D88	D10	D10	D30	D10	D79	D10	D63
D11	D90	D90	D11	D11	D31	D11	D80	D11	D64
D12	D91	D91	D12	D12	D36	D12	D81	D12	D65
D13	D92	D92	D13	D13	D37	D13	D82	D13	D69
D16	D94	D94	D16	D16	D41	D16	D83	D16	D73
D21	D95	D95	D21	D21	D49	D21	D84	D21	D75
D23	D96	D96	D23	D23	D57	D23	D85	D23	D77
D27		D27		D48		D48		D48	
D28		D28		D54		D54		D54	
D29		D29		D87		D87		D87	
D30		D30		D88		D88		D88	
D31		D31		D90		D90		D90	
D36		D36		D91		D91		D91	
D37		D37		D92		D92		D92	
D41		D41		D94		D94		D94	
D44		D44		D95		D95		D95	
D45		D45		D96		D96		D96	
D49		D49		D58		D27		D27	
D57		D57		D61		D28		D28	
D58		D58		D62		D29		D29	
D61		D61		D63		D30		D30	
D62		D62		D64		D31		D31	
D63		D63		D65		D36		D36	
D64		D64		D69		D37		D37	
D65		D65		D73		D41		D41	
D69		D69		D75		D49		D49	
D73		D73		D77		D57		D57	
D75		D75		D44		D58		D44	
D77		D77		D45		D61		D45	
D78		D78		D78		D62		D78	
D79		D79		D79		D63		D79	
D80		D80		D80		D64		D80	
D81		D81		D81		D65		D81	
D82		D82		D82		D69		D82	
D83		D83		D83		D73		D83	
D84		D84		D84		D75		D84	
D85		D85		D85		D77		D85	

SL344. 2021-*-*, * leidyb. apsk. l. Tiražas 16 egz. Užsakymas * .
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas