Training algorithm for speaker-independent voice recognition systems using HTK

K. Nikalaenka [1], Y. Hetsevich [1]
1) United Institute of Informatics Problems, Minsk, Belarus

Abstract: *This paper presents the training algorithm by means of which everyone can develop their own speaker-independent voice recognition system. HTK toolkit is chosen as main tool for recognition process. Through this algorithm user may create voice recognition systems in a short period of time and automatically. Although a usershould prepare input data for training the recognition system. Test recognition systems, working on developed algorithm, have appeared to reach decent accuracy for the Belarusian language and show up themselves viable.*

***Keywords*:** HTK, speech recognition, the Belarussian language, algorithm.

Introduction

One of the most effective and simple means of interaction between people is speech. Speech Processing is based on natural speech interfaces. Natural speech processing comprises two main large areas of knowledge. The first of them – synthesis – gives computer an ability to "speak". Using speaker's voice, computer may report some mathematical results of work, respond to any speaker, voice text data by headphones or speakers. Second field of knowledge is speech recognition. This is inverted function to speech synthesis. Speech recognition helps computer to "understand" what the user is talking about.

Thus, speech recognition is a technology, which allows some technical devices an ability "to understand" text data (audio and voice commands) in defined input audio format. The main purpose of speech recognition is to transform voiced command into text or any other format, which will be simple in understanding to technical device or the user. While solving the general problem of speech recognition some smaller tasks may be set.

- Voice recording and it's digitizing;
- Primary analysis of speech signal;
- Recognition of received voice message;

The main part of each speech technology is called "engine" or the core of the program – a set of data and rules by which data processing will be done. Depending on that core two different types may be extracted: TTS (Text-to-Speech) and ASR (Automatic Speech Recognition). TTS engine realizes speech synthesis, when ASR engine is designed for speech recognition. There are some major developers that create ASR cores: Sphinx, HTK, Julius, Kaldi and others. Some of them are described below.

CMU Sphinx consists of a series of speech recognizers and acoustic model trainer. Sphinx is speaker-independent continuous speech recognizer, which uses hidden Markov models and the n-gram statistical language model [1].

HTK is a toolkit for speech recognition, which uses hidden Markov models. HTK package was developed for processing HMM models. HTK is a set of libraries and tools that can be used in the analysis and speech signals work [2].

Julius — this is continuous large vocabulary speech decoder for research in continuous speech. Forworking with Julious, language and acoustic models should be chosen. Julius adapts acoustic model of HTK ASCII (encoded format), the pronunciation database in HTK format, and 3-level and 2-level-gram language model [3].

Kaldiis similar to HTK in terms of the purpose and field of product's usage. The main goal of the developers is to create a modern and easily portable code that will be easy to modify and expand [4].

There are another, more specific speech recognition systems, such as iATROS, RWTH ASR, Simon, and some slower cloud services like Google ASR and Yandex ASR.

The main purpose of this article is to create service for automatic building of Belarusian speech recognition systems. To do this, it needs to develop first of all an algorithm of the service,then acoustic data for testing purposes, and after all, a prototype with all the functions.

1. HTK AS A TOOL FOR SPEECH RECOGNITION

The Hidden Markov Model Toolkit (HTK) is a portable toolkit for building and manipulating hidden Markov models. HTK is primary used for speech recognition research, although it can be used for numerous other applications including speech synthesis, character recognition and DNA sequencing.

HTK Package is free and may be downloaded on official htk site. HTK is simple in transferring between different platforms. At the same time, it is in use in numerous sites worldwide. HTK comprises a set of library's modules and tools available in C source form The tools provide sophisticated facilities for speech analysis, HMM training, testing and results analysis.

HTK was originally developed at the Machine Intelligence Laboratory (formerly known as the Speech Vision and Robotics Group) of the Cambridge University Engineering Department (CUED) where it has been used to build CUED's large vocabulary speech recognition systems (see CUED HTK LVR). In 1993 Entropic Research Laboratory Inc. acquired the rights to sell HTK. The development of HTK was fully transferred to Entropic in 1995 when the Entropic Cambridge Research Laboratory Ltd was established. HTK was sold by Entropic until 1999 when Microsoft bought Entropic. Microsoft has now licensed HTK back to CUED and is providing support so that CUED can redistribute HTK and provide development support via the HTK3 web site.

After installing in the directory, HTK software package represents a list of executive functions, which may be integrated to different platforms or programming language in the future. Also, these functions can be invoked from the command line to simplify the work with them.

Executable files to work with HTK for Windows

platform may be found on the fig 1.

The main advantages of HTK are:

- High accuracy of recognition;
- Clear representation of the speech signal;
- Strong mathematical tool;
- Effective modeling both temporal and spectral variations of the speech signal;
- Flexible topology;

The main disadvantages of HTK are:

- Weak discriminant power;
- Difficult mathematical tool;
- Huge memory needs for storing parameters of the model and study data;
- Model of the first order, it means state at time n depends on the previous state at n-1time;
- Education and optimization of linguistic model is separated from the acoustic models.

In early 90's, the Markov method was supplemented by neural networks, which essentially complemented the HMM. Thus, hybrid model has been created, which is combining the advantages of both approaches. This model has presented the possibility of simulating long-term dependency on hidden Markov models, and the neural network method has provided universal non parametric approximation, probability estimation, reduction of some parameters for evaluation which are typically required in conventional hidden Markov models.

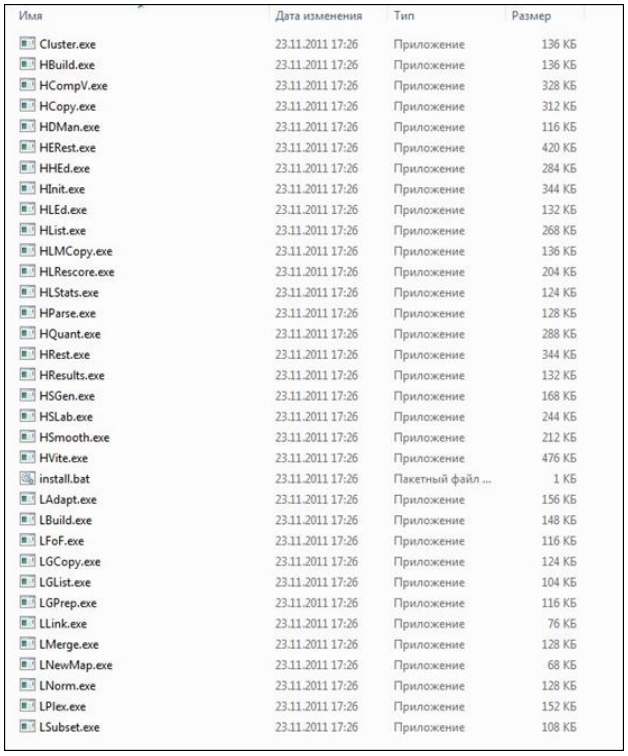

Figure1 – A set of HTK libraries and tools

2. STRUCTURE OF SPEECH RECOGNITION SYSTEMS USING HTK

Speech recognition systems, built on HTK, work in two steps – training and recognition [6].

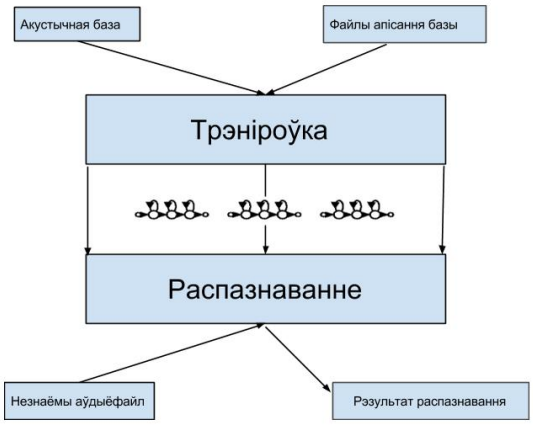

Figure 2 – The algorithm of recognition system based on HTK

For training the speech recognition system the user need to build and complement acoustic base and additional files, describing this acoustic database.

In our case, in the field of laboratory research students were asked to read and record the commands from a limited domain: clothes and footwear.

The domain itself includes 42 commands. Each command is an element of clothing or footwear in the Belarussian language. Commands were recorded in alphabetical order. The most number of voices in acoustic base are women's voices.

3. INPUT AND OUTPUT DATA

In any actual and working speech recognition system input data are submitted as text or audio files. But, when it comes to speech recognition service development, first of all, before starting the process, the possibility of automatic creation of such systems should be developed. Such service should take as input data, which will form the base and describe future recognition system. In our case, format of input data consists of name of the command, which system is able to recognize, path to the file in acoustic base, phonemic composition, addition data for speech synthesis. There are some examples of input commands in defined format:

1 басаножкі
cache/windows/in/wavs/input/db_vopratka_I_elemienty/Hanna/1.wav B,A,S,A,N,O,SH,K',I
B004,A312,S002,A222,N002,O022,SH002,K'002,I340
B004(122ms;8000hz),A312(91ms;8000hz),S002(161ms;8000hz),A222(121ms;8000hz),N002(155ms;8000hz),O022(224ms;8000hz),SH002(171ms;8000hz),K'002(164ms;8000hz),I340(150ms;8000hz)

2 блуза
cache/windows/in/wavs/input/db_vopratka_I_elemienty/Hanna/2.wav B,L,U,Z,A B001,L002,U022,Z004,A320
B001(130ms;8000hz),L002(128ms;8000hz),U022(223ms;8000hz),Z004(105ms;8000hz),A320(150ms;8000hz)

3 боты
cache/windows/in/wavs/input/db_vopratka_I_elemienty/Hanna/3.wav B,O,T,Y B002,O012,T002,Y320
B002(134ms;8000hz),O012(194ms;8000hz),T002(140ms;8000hz),Y320(150ms;8000hz)

4 гальштук

cache/windows/in/wavs/input/db_vopratka_I_elemienty/Hanna/4.wav GH,A,L',SH,T,U,K
GH002,A033,L'003,SH002,T002,U322,K000
GH002(142ms;8000hz),A033(224ms;8000hz),L'003(135ms;8000hz),SH002(171ms;8000hz),T002(140ms;8000hz),U322(91ms;8000hz),K000(159ms;8000hz)

5 гузік
cache/windows/in/wavs/input/db_vopratka_I_elemienty/Hanna/5.wav GH,U,Z',I,K GH001,U033,Z'004,I342,K000
GH001(134ms;8000hz),U033(234ms;8000hz),Z'004(124ms;8000hz),I342(91ms;8000hz),K000(159ms;8000hz)

User has to prepare input data before working. Audio data, which will be used as base in the future, may be recorded by any software with such function, for example Soundforge [5], Windows sound recording or through any online recorder. The format of audio data for developed service is .wav. At the same time, to build speech recognition system, user has to prepare allophones and phonemes of all the commands in system. Such operations may be simply done by using special service of the speech synthesis and recognition laboratory of The United Institute of Informatics Problems NASB – corpus.by [7]

As results the user will receive his own speech recognition system with its own acoustic base, consisting of any needed number of commands. Such system will provide word recognition with some accuracy rate.

4. DESCRIPTION OF THE ALGORITHM FOR AUTOMATIC CREATION OF SPEAKER-ONDEPENDENT SPEECH RECOGNITION SYSTEMS BASED ON HTK

The algorithm consists of 5 steps:

Step 0. *Cleaning*. Due to the fact that speech recognition systems based on this algorithm may be created, deleted and modified very frequently and quickly, the very first step in the algorithm should allow the user to remove all temporary and irrelevant data. In order to preserve old version it makes sense at this stage to save the previous data to another folder called “previous version”, instead of deleting it.

Step 1. Creating all necessary for HTK files. On this step input data for speech recognition system is analyzed and a list of necessary files for both training and recognition steps of HTK system are created.

Step 2. *Training*. On the second step the service starts automatic training of the system. All files, created on the previous step, are processed by HTK functions. During this process additional HTK files are created and training is carried out.. After that step, the user will receive complete speech recognition system, ready for work and tests. This step may be considered as first from two main phases of speech recognition systems.

Step 3. *System testing on input data.* This step carries speech recognition system testing on audio data that were used while training earlier. The main purpose of this step is quality control of the system on its "native" audio files. In theory, an accuracy of recognition process should be close to 100%. Additional purpose of that step may be considered as economy of the user’s time during huge acoustic base processing. The process of testing should be automatic, or in other case it will become impossible to check and modify acoustic base in the future.

Step 4. *Speech recognition.* The last step of the developed algorithm is speech recognition. The user can make speech recognition using received speech recognition system of any audio data of restricted format. As the result, one command from the list of possible commands will be recognized with some accuracy. Quality of speech recognition depends on many factors.

5. PRACTICAL USAGE OF THE ALGORITHM

During the research a service-prototype for creation speech recognition systems was developed. Such service allows the user to set up properly working speech recognition system in few simple steps, using developed algorithm. The prototype was generated entirely in PHP programming language. A lot of parsers and scripts were developed for HTK software package. They help to use HTK functions automatically. On their base all necessary linguistic and acoustic resources may be created. External service interface is shown on figure 3.

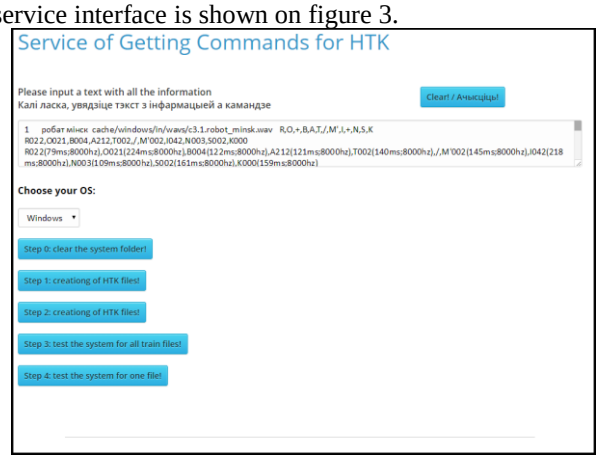

Figure 3 – External interface of developed prototypehe

Using this online service the user receives a finished speech recognition system, that can be tested on any audio file from defined format. The example of the service operation is shown on figure 4:

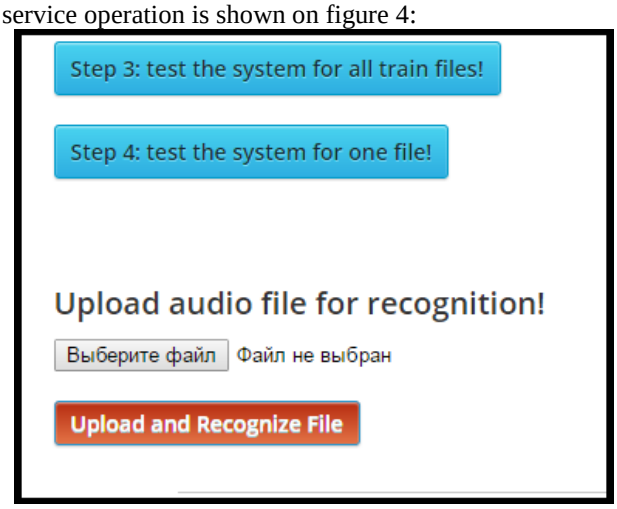

Figure 4 – Example of work of the prototype

The results of the recognition are shown on figure 5. The list of recognized commands is added to list, which is shown on the monitor. At the same time the user can get links to audio files and text files, which describe results of the recognition.

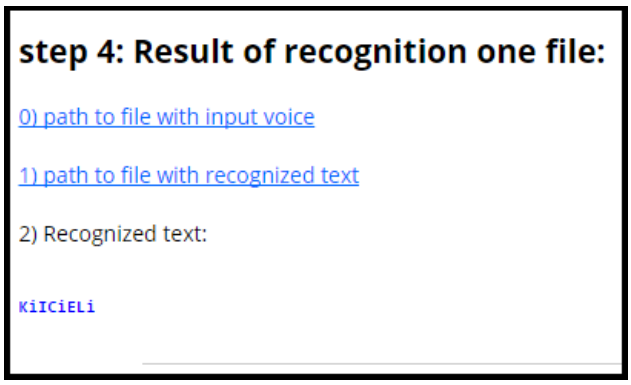

Figure 5 – Result of speech recognition

As it is shown on figure 5, command “Kicel” was recognized.

6. RECOGNITION TESTS

After implementing this prototype, some tests were made..

Acoustic base used for tests consisted from:

– Number of commands in single domain – 42;

– Number of speakers - 32;

– Number of commands at all - 1383.

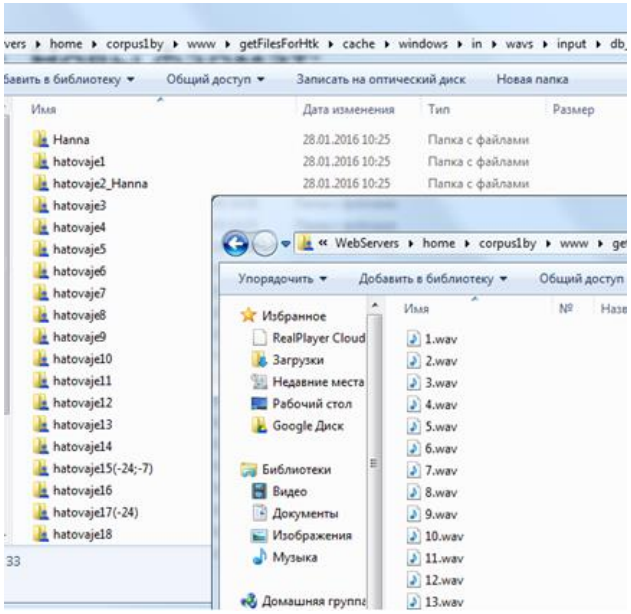

Figure 6 – Acoustic base used for speech recognition training

A set of tests was made. As a result, accuracy appeared to be dependent on the parameter “size of acoustic base”, on which speech recognition system was trained. Moreover, if the same speaker was recorded multiple times, It would slightly increase the quality of recognition aswell. However, besides the quality there is a parameter called speed of processing. Our automatic creation of speech recognition system makes the process of recognition faster then manual one in more then 5 times. Average rate of recognition may reach 4200 commands per hour, what is more then 3 times faster then doing the same recognition manually. The best accuracy what was reached during tests on the whole acoustic base of 1383 commands is – 56.6%. The best accuracy on smaller base is up to 92.2% when acoustic base consisted of only 460 commands.

7. CONCLUSION

As a result of this paper speech recognition software was selected and tested. The main advantages and disadvantages of HTK were analyzed. A prototype of service for automatic creation of speech recognition systems was developed. Service allows user to make his own speech recognition system of any size automatically using HTK.

Some tests of developed service were made. The main 2 goals of tests were – accuracy of speech recognition systems and their speed. As the result, the best accuracy which was received stays near 92%, but if user will put in much bigger then 500 commands acoustic base, accuracy may go lower up to 50% in such speech recognition systems. Such difference may be caused by few factors like: quality of acoustic base or other mathematical parameters, which describe the base (number of training circles, marking and etc.).

8. REFERENCES

[1] CMU Sphinx [Electronic resource] Mode of access: http://cmusphinx.sourceforge.net/ - Date of access: 15.05.2016.

[2] HTK [Electronic resource] Mode of access: http://cmusphinx.sourceforge.net/ - Date of access: 15.05.2016.

[3] Julius [Electronic resource] Mode of access: http://julius.sourceforge.jp/en_index.php/ - Date of access: 15.05.2016.

[4] KALDI [Electronic resource] Mode of access: http://kaldi.sourceforge.net/ - Date of access: 15.05.2016.

[5] SoundForge [Electronic resource] Mode of access: http://sonycreativesoftware.com/audiostudio/ - Date of access: 15.05.2016.

[6] Нікалаенка, К.А. Кампаненты для розных платформаў сінтэзатара маўлення па тэксце для інтэлектуальных сістэм / К.А. Нікалаенка, Л.І. Кайгародава, Ю.С. Гецэвіч // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS–2015) : материалы V Междунар. науч.-техн. конф. (Minsk, 21–23 February 2015 year) / редкол. : В.В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2015 г. – p. 507–512.)

[7] Corpus.by [Electronic resource] Mode of access: http://www.corpus.by - Date of access: 15.05.2016.