IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. IV (Sep.- Oct. 2017), PP 71-79
www.iosrjournals.org

Speech Recognition Systems: A Comparative Review

Rami Matarneh[1], Svitlana Maksymova[2], Vyacheslav V. Lyashenko[3]
, Nataliya V. Belova [3]
[1](*Department of Computer Science, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabi)*
[2](*Department of Computer-Integrated Technologies, Automation and Mechatronics, Kharkiv National University of RadioElectronics, Kharkiv, Ukraine)*
[3](*Department of Informatics, Kharkiv National University of RadioElectronics, Kharkiv, Ukraine)*

Abstract: *Creating voice control for robots is very important and difficult task. Therefore, we consider different systems of speech recognition. We divided them into two main classes: (1) open-source and (2) close-source code. As close-source software the following were selected: Dragon Mobile SDK, Google Speech Recognition API, Siri, Yandex SpeechKit and Microsoft Speech API. While the following were selected as open-source software: CMU Sphinx, Kaldi, Julius, HTK, iAtros, RWTH ASR and Simon. The comparison mainly based on accuracy, API, performance, speed in real-time, response time and compatibility. the variety of comparison axes allow us to make detailed description of the differences and similarities, which in turn enabled us to adopt a careful decision to choose the appropriate system depending on our need.*
Keywords: *Robot, Speech recognition, Voice systems with closed source code, Voice systems with open source code*

Date of Submission: 13-10-2017 Date of acceptance: 27-10-2017

I. Introduction

Voice control will make your application more convenient for user especially if a person works with it on the go or his hands are busy. Without touching the screen, it can call the desired function in one phrase. In [1] Anna Caute and Celia Woolf propose to use speech recognition technologies for an individual with severe acquired dysgraphia Robert Godwin-Jones [2] analyzes speech recognition technologies for language learning. In [3] authors analyze current speech recognition technologies in order to provide real-time voice-base machine translation, especially Micrsoft Speech API. Dragoș Ciobanu [4] thinks researches in the field of automatic speech recognition are very perspective. Voice control can be also used in different fields of industry especially in those fields where robots using are widely spread. Using such control allows to achieve next results:

– Worker tiredness decreases.
– Commands transmission speed and flexibility increases.
– Hands are freed to perform other functions (for example, to record the flow of the process).
– More saturated information is transmitted in response to the situation that has arisen;
– Invalids labor activity begins.
– Work monotony is reduced, since the operator can use his own hearing to monitor the accuracy of the commands being submitted, thereby becoming more actively involved in the workflow.

Solving these problems requires more detailed consideration of the voice systems that exist.

II. Literature Review

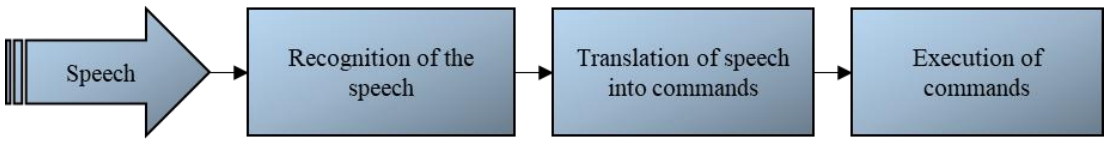
In [5] Kuldeep Kumar, R.K. Aggarwal, Ankita Jain create their own speech recognition system using hidden Markov model toolkit (HTK) and Mel frequency cepstral coefficient (MFCC) method. Authors analyze current speech recognition technologies in order to provide real-time voice-base machine translation. Hanna Suominen, Liyuan Zhou, Leif Hanlen, Gabriela Ferraro [6] propose to use speech recognition to prevent failures in information flow in health care, they use Dragon Medical 11.0 for this task. In [7] analyzed cloud-based speech recognition systems using Siri, Google Speech Recognizer and Dragon. A lot of authors compare systems between themselves. Belenko M.V., Balakshin P.V. [8] analyze systems with open source code, enter evaluation coefficient for different parameters and make recommendations for recognition systems using. HTK and Julius are recommended for use in educational activities in speech recognition field. Kaldi can be successfully applied for research activities [8]. In [9] analyze deep neural networks (DNNs) with many hidden layers using and their training using different methods. In [10] authors use Deep Belief Networks to pretrain context-dependent artificial neuron net (ANN) / HMM system trained on two datasets. Google speech

recognition engine was modified to incorporate a neural network frontend. In [11] authors try to use Google Speech Recognition for under-resourced languages. Authors of [12] from Google Inc. report the development of an accurate, small-footprint, large vocabulary speech recognizer for mobile devices. Mohit Dua, R.K.Aggarwal, Virender Kadyan, Shelza Dua [13] use HTK for Punjabi Automatic Speech Recognition. In [14] authors use CMU Sphinx for Arabic phonemes. Authors of [15] use convolutional neural networks for error rate reduction. Jinyu Li, Li Deng, Yifan Gong, Reinhold Haeb-Umbach [16] analyze techniques for development of noise-robust speech recognition systems. Oliver Lemon [17] analyzes speech recognition interfaces, so do Jerome R. Bellegarda [18] and Li Deng, Xiao Li [19] from these papers we can distinguish Siri. Silnov Dmitry Sergeevich [20] use Google Speech and Yandex SpeechKit for decoding radio talks. In [21] authors propose their recognition system based on CMU Sphinx. Authors of [22] propose to use Microsoft Speech API for home automation. Howard Hao-Jan Chen [23] discusses how Microsoft Speech SDK can be used to develop an oral skills training website for students. Povey, D. el. [24] describe Kaldi Speech Recognition Toolkit. Ivan Tashev [25] and R. Maskeliunas, K. Ratkevicius, V. Rudzionis [26] analyze using Microsoft Speech Engine for human-machine interaction. Y. Bala Krishna, S. Nagendram [27] and Faisal Baig, Saira Beg and Muhammad Fahad Khan [28] propose to use Microsoft Speech API for smart home appliances. In [29] authors propose to use Microsoft Speech API for the development of an assistive technology to provide a solution for communication between two physically disabled persons; blind and deaf. In [30] authors use Microsoft Speech API for evaluation response on audience. Authors [31] develop examination system with the use of Microsoft Speech API that can be also used for the students with disabilities. In [32] authors also analyze using Microsoft Speech API for question answering systems. CMU Sphinx tools were used in [33] for training and evaluation a language model on Holy Quran recitations. In [34] authors also use CMU Sphinx to train system and decode speech data. Hassan Satori, Fatima ElHaoussi [35] develop their own speaker-independent continuous automatic speech recognition system using Sphinx tools. The same authors in [36] prove they can define Smokers and Non-Smokers using their system based on CMU Sphinx. In [37] authors also use Sphinx to generate subtitles via a three staged process: Audio extraction, Speech Recognition and Synchronization of subtitles. In [38] CMU Sphinx was used in order to recognize Polish speech. Medennikov I., Prudnikov A. [39] Kaldi speech recognition toolkit was used for experiments with Russian speech recognition. Authors of [40] use Kaldi for African speech recognition. In [41] authors use Kaldi for speech recognition. Authors [42] try to develop Kaldi speech recognition system compatible with Julius. In [43] authors use Julius for speech recognition system development for robots. For mobile applications authors [44] use Julius. Authors of [45] use HTK for recognition whisper. In [46] HTK was used for Telugu language recognition. And in [47] HTK was used for their own speech-to-text system.

III. Variety Of Voice Systems

Voice control implementation may be conditionally divided into parts: speech, recognition, translation, and execution of commands (Fig. 1).

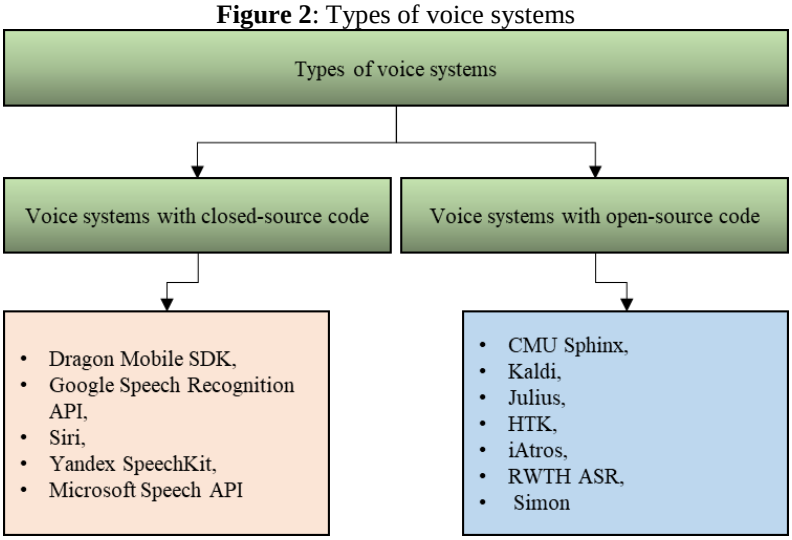
Figure 1: Voice control implementation



IV. Voice Systems With Closed And Open Source Code

All speech recognition engines work similarly, where the user's voice is passed through microphone to reach the recognition system. Which is mean that we have two general algorithms. The first: voice is processed on local device and the second: record is sent to the remote server for further processing. The second variant is more suitable for smartphones and tablets. So do commercial engines Cortana, Google Now and Siri. At the same time, we have to divide speech recognition systems into systems with closed and open source code (Fig. 2) [1].

Figure 2: Types of voice systems



V. Voice Systems With Closed Source Code

Closed source code also called proprietary software. This means there is no access to program source code and the software shared as only binary version to prevent any modifications, while access to source code is granted when signing a non-disclosure agreement.

Speech recognition software development world leader is Nuance Communications. Its product Dragon Mobile SDK [2,4] consists of client and server components. It also includes different code examples, templates, documentation and framework. These components simplify services integration to applications. This product uses platform Speech Kit which is also developed by Nuance Communications. The platform allows to add speech recognition and synthesis (TTS, Text-to-Speech) services to projects and applications quickly, it also provides access to speech processing components located on server through asynchronous "pure" net Application Programming Interfaces (API). Thereby reducing the consumed resources and minimize costs. Servers system provides most speech processing operations, so that speech recognition and synthesis is fully implemented on server.

Speech Kit is a net service, it requires base settings in order to use recognition and synthesis classes. First of all, it is necessary to identify and authorize the application and then to establish a connection with the speech processing server. All of the above provide quick queries creation for speech information processing and improves work quality.

Platform performs next agreed processes.

1- Full audio system control for record and play.
2- The network component controls connections to the server and automatically recovers the connection with the elapsed timeout for each new request.
3- End speech detector determines when the user stop talking and if it is necessary it stops record automatically.
4- Coding component compresses and decompresses audio streaming. It decreases requirements to bandwidth and decreases average delay time.

Recognition technology allows user to dictate instead of typing when it is necessary to input the text. Speech recognizer gives text results list. It is not attached to any User Interface (UI) object. That is why most suitable result selection and alternative results selection depends on each application user interface.

Dragon Mobile SDK has high recognition accuracy in English, up to 99%. The main disadvantage is a limited free functionality not more 10000 requests per day, above ten thousand - paid access. Google Speech Recognition API is a technology widely spread in different research fields and for different languages [7, 10-12, 20]. It is a Google's company product. It allows to use voice search on the basis of speech recognition theory. This technology is integrated into smartphones and computers with speech recognition opportunity. In summer 2011 Google integrated speech technology into its search system (Google Search). On personal computers (PC) this technology is supported only by Google Chrome browser. It is also supported by smartphones with Android operation system.

At first Google Voice Search supported only short request with length up to 35-40 words. For send request, it was necessary to turn the microphone on and off. This function is still in Google Search bar; you just need to press microphone to begin. But in 2013, February, continuous speech recognition possibility was added into Chrome, thus Google Voice Search transformed into Speech input.

From 2014, May, access to API became possible. In order to work with database, it is necessary to register in Google Developers. Google Voice Search is introduced to plenty of popular services: Google, YouTube, Yahoo, DuckDuckGo, Bing, Wolfram|Alpha, Wikipedia etc…. There is an opportunity to add own search systems. Now there is even an extension that adds voice information input button for sites using search forms HTML5. And it is necessary to use a microphone for application work.

In order to use Google Voice Search technology it is necessary to perform POST-request for an address with audio data formatted «.flac» or «.spx». Then WAVE-files must be recognized by any application. Goggle Speech Recognition API is mostly like Dragon Mobile SDK by Nuance, but has no limitations on the requests number per day. Google developers use deep neural net for key phrase "Okay, Google" recognition. Linux Speech Recognition is also based on Google Voice API.

Siri [7, 16-19] – Speech Interpretation and Recognition Interface - is an intelligent personal assistant, part of Apple Inc.'s iOS, watchOS, macOS, and tvOS operating systems. The assistant first appeared on the iPhone 4S and it was described by Apple as the best thing on the iPhone during the launch presentation. Siri has access to every other built-in application on your Apple device - Mail, Contacts, Messages, Maps, Safari, etc - and will call upon those apps to present data or search through their databases whenever she needs to. Ultimately, Siri does all legwork for you. It means you can carry out a single task by just saying "Hey Siri" or double tapping the Home button, rather than open multiple apps or spend time writing messages or finding contacts. Siri supports English (Australia, Canada, India, New Zealand, Singapore, UK, US), Spanish, French, German, Italian, Japanese, Korean, Mandarin, Norwegian, Cantonese, Swedish, Danish, Dutch, Russian, Turkish, Thai and Portuguese.

Siri assistant learns to recognize speech using users' queries. Whenever Siri is turned on in the settings, the system warns Apple reserves the right to store and process everything you say. Every time you speak with a voice assistant, it sends data to Apple's data centers for analysis. Apple forms a random sequence of digits and assigns it to the user, and then associates with it all voice query files and their textual decryptions.

Yandex SpeechKit is also widely used, especially for Russian speech recognition [20]. Developers assure this SDK is best choice for Russian language using. There is a limitation not more 10000 requests per day. First of all, recognition effectiveness in Yandex Speech Kit depends on original sound quality, coding, speech intelligibility, speech tempo, complexity and phrases length. Voice requests subject should be the same as the selected language model, this definitely will increase the accuracy of recognition. Speech recognition is performed in real time with audio information transmission. Recognition speed depends on audio data transmission method. If data is transmitted in parts, recognition is performed simultaneously with data transfer. The delay does not exceed one second. The technology works in the streaming recognition mode with intermediate results in order to provide such high speed. So when the user begins speaking, his speech is transmitted to the recognition service in small parts, where the SpeechKit Cloud converts the received audio data into mono PCM / 16 bit / 16 kHz.

Yandex speech technologies includes recognition, synthesis speech, voice activation and highlighting the semantic objects in the spoken text. SpeechKit Cloud is the interface to access recognition and synthesis speech technologies. It designed taking into account high loads in order to provide the availability and system trouble-free operation, even with large number of simultaneous requests. Interaction with SpeechKit Cloud is performed through HTTP API. So we can implement different functions in short time.

Yandex speech technology supports computer games and applications, voice control in the car, interactive voice menu (Interactive Voice Response) in telephony, voice interface for "Smart house", electronic robots voice interface, voice control for home appliances etc.

To get acquainted with the technology, there is a free test period: one month from the moment of sending the first request to the server, in order to use SpeechKit Cloud after this month it is necessary to enter into contract with two distinct choices: either buy a package (fixed number of requests per month) or pay for requests and the cost will depend on the number of request.

SpeechKit Cloud now supports English, Russian, Ukrainian and Turkish languages. It recognizes speech through two stages: At the first stage, it allocates sound sets in the audio signal, which can be interpreted as words) for every sound set there are several words variants – hypotheses (. At the second stage, language model is taken into account, which allows to check each hypothesis from the point of view of language structure and context (how much this word is consistent with the words recognized earlier). Recognition system checks hypotheses using language model as a vocabulary. This vocabulary uses neuron nets machine learning. Neuron net is taught using speech used in certain field. That is why language models are specialized for certain topic speech recognition. To prepare models, large data sets are used from Yandex services and applications.

Let us take a deep look at recognition module construction of Yandex speech technology. Recorded speech stream is divided into 20 ms frames, the signal spectrum is scaled and after transformations the MFCC is obtained for each frame. These coefficients enter acoustic model to calculate the probability distribution for

approximately 4000 senons in each frame, where senon is phoneme's beginning, middle or end. SpeechKit acoustic model is constructed using a combination of hidden Markov models (HMM) and a feedforward deep neural network (DNN).

Then comes the first language model: several weighted finite transducers (WFST) convert the senons into context-dependent phonemes. Whole words are constructed using the pronunciation vocabulary from them, then hundreds of hypotheses are obtained for each word. Final processing is performed in the second language model; it uses recurrent neural net (RNN), which ranks the received hypotheses to choose the most plausible variant by defining each word context to be able to take into account the influence of the nearest words and the further parts as well. Long coherent texts recognition is available in SpeechKit Cloud and SpeechKit Mobile SDK. To use the new language model in the query parameters, select the topic "notes".

Voice interface the second key component is voice activation system. It starts the desired action in response to the key phrase, its technology is rather flexible, the developer using SpeechKit can choose any key phrase for his application.

Here the difference between Goggle and Yandex is located. DNN gives high quality but activation system is limited only by one key phrase and for its training a massive of data is necessary.

SpeechKit Box allows to implement speech recognition, synthesis functions and semantic analysis of everything spoken in services and applications. This complex is deployed on client's internal network, so that the data is not transferred for processing to external servers. Due to this, speech technologies can be used to work with confidential information.

Microsoft Speech API is the product of Microsoft Company which is often used for different tasks [22, 23, 25-32]. This system is mostly like GoogleSpeech API and YandexSpeech AP with few differences. Microsoft Speech Application Programming Interface (SAPI) with Microsoft Speech SDK is used for voice commands processing application development. This API includes a set of effective methods and data and well integrated into .NET framework providing a new development platform accessible to personal computers. At last, it works with several automatic speech recognition methods; because they give certain freedom for developers to select technology and speech processing mechanism.

Microsoft voice assistant Cortana was announced and it was also announced development of automatic technology of synchronous tele-translation from English to German and vice versa for Skype.

Now it can be used in four variants: (1) for Windows application, Speech Engine can be added using controlled and native code which can be got from API to control Speech Engine embedded in Windows and Windows Server, (2) Speech Platform may be embedded into applications using Microsoft's distributives (language packages with speech recognition functions or tools for TTS), (3) embedded solutions that allow human to interact with devices using voice commands (e.g. Ford cars control using voice commands in OS Windows Automotive) and (4) application with speech functions development can be used in real time, it frees human from development, service and modernization speech solutions infrastructure.

VI. Voice Systems With Open Source Code

Now let us consider speech recognition systems with open source code. One of the most famous systems is CMU Sphinx [14, 21, 33-38, 48] or simply Sphinx. It was mainly written by the Carnegie Mellon University speech recognition developers group – Xuedong Huang. It includes speech recognizers series (Sphinx 2-4) and acoustic model trainer (Sphinx train). In 2000, Carnegie Mellon's Sphinx group approved open source speech recognition system components including Sphinx 2 and later Sphinx 3 (in 2001). For SourceForge Kevin Lenzo in Linux World in 2000 it was released in Open Source on the basis of BSD-license (Berkeley Software Distribution – software distribution system in source codes developed for the exchange of experience between educational institutions). In brief, this license may be characterized as follows: all source code belongs to BSD and all corrections belong to their authors. Speech decoder included acoustic models and simple applications, while the available resources also include additional software for acoustic model training, language model, linguistic compilation model and a pronunciation vocabulary. Sphinx is a speaker-independent continuous speech recognizer was developed by Kai-Fu Lee and uses HMM and n-gram statistical language model, which is able to recognize continuous speech recognition with speaker-independent big vocabulary. Sphinx in its historical development eclipsed all previous versions in terms of its performance where Sphinx2 is the fastest and performance-oriented speech recognizer.

Sphinx2 uses dialog system language learning system and it is oriented on speech recognition in real time which makes it ideally suited for developing various mobile applications. It includes such functionality as the final pointer, partial generation of hypotheses, dynamic language model connection and so on. Sphinx2 code was incorporated into numerous commercial products, but it was not developed actively for a long time. Sphinx3 represents semi continuous speech recognition acoustic model, adopted a common continuous model constructed on HMM. It was originally used for high-precision speech recognition, which was carried out in the post-factum mode. Last developments (algorithms and software) allows Sphinx to recognize in a mode close to

real-time, but it was not suitable yet for high-quality use as an application. After active development and reunification with SphinxTrain, Sphinx3 provided access to numerous modern techniques and models such as LDA/MLLT, MLLR and VTLN that improved speech recognition accuracy. Now Sphinx is not a user application, it is rather a toolkit that can be used in order to develop applications for end users with a powerful capability for speech recognition. It includes several parts:

1- PocketSphinx is small fast program, processing sound, acoustic models, grammars and dictionaries.
2- library Sphinxbase is necessary for PocketSphinx work.
3- Sphinx4 is recognition library.
4- Sphinxtrain is a software for acoustic models training.

According to [8] we see this system shows a medium recognition accuracy and the highest speed, taking into consideration that the use of PocketSphinx would significantly increase the speed. PocketSphinx can be used with different platforms including Android, it is also well integrated into projects written in Java, in addition to its module structure allows making changes and fixing bugs quickly, moreover except console it provides API with all its advantages and it has detailed documentation. By default, this system supports plenty of languages, i.e. it contains free access to language and acoustic models for these languages.

Kaldi provides a lot of modern approaches currently used in speech recognition [24, 39-42], which is allow using a variety of algorithms to reduce the acoustic signal characteristics size to increase system's performance. Kaldi is written in C++ and as Sphinx it also has module structure, so new functions can be added easily and errors can be corrected quickly. This system supports different platforms, but it provides only console that complicates its integration to other applications.

By default, Kaldi supports only English language and provides detailed documentation which is oriented only to experienced readers in speech recognition field. It is distributed under the fully free Apache license that is, it can be integrated into a commercial product without disclosing its code According to [8] it shows the best recognition accuracy and high recognition speed and has leading algorithms and data structures.

Julius [1, 42-44] was developed as a free software part for Japanese language researches. From version 3.4, grammar analyzer got name Julian, which uses its own grammar on the base of finite state machine and was integrated in Julius. Julian also a secure modulation and it may be independent of model structures and HMM different types. It has open source and is distributed with a BSD license type.

Julius is a large vocabulary continuous speech recognizer with software decoder for research in coherent speech field, which makes it ideal for decoding in near real-time mode on most existing computers. It has 60000 words in vocabulary and it also uses HMM. The main of its features is the full embedding. Its base platform is Linux and other UNIX-like systems, but version for Windows also exists.

In order to use Julius, it is necessary to select language and acoustic model for specified language. Julius adapts the acoustic model of the encoding format HTK ASCII, HTK format pronunciation database and 3-layer diagram of the ARPA standard language model construction. The main Julius disadvantage is usefulness only for Japanese language, although VoxForge project is working over acoustic model for English language using recognition system engine.

Acoustic and language stages are performed using some utilities included in HTK based on Viterbi algorithm. This system is implemented using C language and provides console and API for integration into third-party applications. From [8] this system shows worst accuracy rate and medium recognition speed.

HTK is also widely spread [13, 45-47, 49]. It is implemented in the C language which leads to increases work speed; because C is low-level programming language. By its structure this system is divided into utilities set which can be called from the command line and it also provides API named ATK. By default, it supports only English language and distributed under the HTK license, which allows the distribution of the system source code. It has HTK Book that describes HTK work and speech recognition systems work main principles [13].

From [8] we can make a conclusion that this system has mediocre results. It provides only classical speech recognition algorithms and data structures, while it is easy to use along with Julius.

iAtros is implemented in C and also has module structures with all its advantages, but has not a lot of algorithms and data structures, except speech recognition functions it contains text recognition module. It is not convenient and does not provide API for easy integration into third-party applications. By default, it supports English and Spain languages. It is not cross-platform, as it is only run under the operating systems of the Linux family and distributed under GPLv3 that does not allow embedding system into commercial projects without disclosing their source code, so it cannot be used in commercial activity. In addition, it can be used if except speech pattern recognition is necessary. According to [8] recognition accuracy is rather good but speed is rather low.

RWTH ASR (briefly RASR) [24, 48] is an open-source speech recognition tool. It includes speech recognition technology for automatic speech recognition system development. This technology is developed by the Natural Language Technology Center and the Model recognition group at the Rhine-Westphalian Technical

University of Aachen. RWTH ASR includes tools for acoustic models development and decoders and also components for speaker speech adaptation, uncontrolled learning systems, differential learning systems and lattice word processing forms. This software works with OS Linux and Mac OS X. Project homepage proposes ready for use models with tasks, teaching systems and documentation.

The toolkit is published under an open source license, called the "RWTH ASR license", which is derived from the QPL license (Q Public License). This license represents free use possibility, including redistribution and modification for non-commercial use. A distinctive feature is the ability to use the voicing characteristic when extracting the acoustic characteristics of the input signal. This system can use weighted finite state machine as a language model in the stage of language modeling. This system is implemented using C++ and also has module structure.

Documentation is not full; because it describes only installation process. By default, it supports only English language and considered as not a cross-platform system and cannot work under Windows, in addition provides only console. From [8] RWTH ASR has not bad recognition accuracy but it has the worst speed.

Simon is speech recognition system based on Julius and HTK engines. It is convenient for working with different languages and dialects. In this case, the speech recognition response is fully customizable and it is not suitable for exclusive recognition of single voice requests and cannot be configured for users' needs.

In order to use it, certain "scripts" must be performed. Among possible scripts, e.g. "Firefox" (running and managing the browser "Firefox") or "window control package" (close, move, dimensions change) etc. Scripts can be created by users and distributed in the community through the Get Hot New Stuff system.

Simon also supports different model like General Public License (GPL) models from Voxforge used by users for English, German, Portuguese pronunciation and there is no need to train system to start working.

GPL purpose is to grant the user the right to copy, modify and distribute (even commercial) software (which is prohibited by copyright law by default). The principle of rights "inheritance" is called copyleft which was founded by Richard Stallman. In comparison with GPL standard licenses for proprietary software, they rarely give the user such rights.

VII. Conclusion

Proceeding from the foregoing we can make a conclusion that from open resource code systems, CMU Sphinx is the most interesting. However, to get effective results a large initial database is necessary, otherwise the recognition accuracy will be modest, especially in comparison with systems with closed source code

Thus, after considering the most common speech recognition systems with closed source code we can say that the most accurate is Dragon NaturallySpeaking; because it is most suitable for recognition tasks, has good documentation and simple API code for embedding. But we must notice that this toolkit has a very complex licensing system. Therefore, it becomes difficult to implement a custom product on the Dragon Mobile SDK. In this case Google Speech API is more convenient. It is more embeddable and fast due to large computing power, in addition to no limitations on requests number per day.

The main advantage of closed-source recognition systems (but open API for developers) when compared with open source speech recognition systems is high accuracy (due to huge database libraries) and speech recognition speed.

References

[1] Caute, A., & Woolf, C. (2016). Using voice recognition software to improve communicative writing and social participation in an individual with severe acquired dysgraphia: An experimental single-case therapy study. *Aphasiology*, *30*(2-3), 245-268.

[2] Godwin-Jones, R. (2011). Mobile apps for language learning. *Language Learning & Technology*, 15(2), 2-11.

[3] Duarte, T., Prikladnicki, R., Calefato, F., & Lanubile, F. (2014). Speech recognition for voice-based machine translation. *IEEE software*, *31*(1), 26-31.

[4] Ciobanu, D. (2014). Of Dragons and Speech Recognition Wizards and Apprentices. *Tradumàtica*, (12), 0524-538.

[5] Kumar, K., Aggarwal, R. K., & Jain, A. (2012). A Hindi speech recognition system for connected words using HTK. *International Journal of Computational Systems Engineering*, *1*(1), 25-32.

[6] Suominen, H., Zhou, L., Hanlen, L., & Ferraro, G. (2015). Benchmarking clinical speech recognition and information extraction: new data, methods, and evaluations. *JMIR medical informatics*, *3*(2), E19.

[7] Assefi, M., Liu, G., Wittit, M. P., & Izurieta, C. (2016). Measuring the Impact of Network Performance on Cloud-Based Speech Recognition Applications. *International Journal of Computer Applications-IJCA*, *23*, 19-28.

[8] Belenko, M.V. & Balakshin, P.V. (2017). Comparative Analysis of Speech Recognition Systems with Open Code. *Mezhdunarodnyiy Nauchno-Issledovatelskiy Zhurnal*, 4 (58), 13-18.

[9] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82-97.

[10] Jaitly, N., Nguyen, P., Senior, A., & Vanhoucke, V. (2012). Application of pretrained deep neural networks to large vocabulary speech recognition. In *Thirteenth Annual Conference of the International Speech Communication Association*, Portland, OR, USA September 9-13, 2578-2581.

[11] Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, *56*, 85-100.

[12] Lei, X., Senior, A. W., Gruenstein, A., & Sorensen, J. (2013, April). Accurate and compact large vocabulary speech recognition on mobile devices. In *Interspeech*, Lyon, France, 1, 662-665.

[13] Dua, M., Aggarwal, R. K., Kadyan, V., & Dua, S. (2012). Punjabi automatic speech recognition using HTK. *IJCSI International Journal of Computer Science Issues*, *9*(4), 1694-0814.

[14] El Amrani, M. Y., Rahman, M. H., Wahiddin, M. R., & Shah, A. (2016). Building CMU Sphinx language model for the Holy Quran using simplified Arabic phonemes. *Egyptian Informatics Journal*, *17*(3), 305-314.

[15] Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, *22*(10), 1533-1545.

[16] Li, J., Deng, L., Gong, Y., & Haeb-Umbach, R. (2014). An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(4), 745-777.

[17] Lemon, O. (2012). Conversational interfaces. In *Data-Driven Methods for Adaptive Spoken Dialogue Systems* (pp. 1-4). Springer New York.

[18] Bellegarda, J. R. (2014). Spoken language understanding for natural interaction: The siri experience. In *Natural Interaction with Robots, Knowbots and Smartphones* (pp. 3-14). Springer, New York, NY.

[19] Deng, L., & Li, X. (2013). Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(5), 1060-1089.

[20] Silnov, D. S. (2016). Special features of radio interception of APCO P25 messages in Russia. *International Journal of Electrical and Computer Engineering*, *6*(3), 1072-1076.

[21] Smirnov, V., Ignatov, D., Gusev, M., Farkhadov, M., Rumyantseva, N., & Farkhadova, M. (2016). A Russian Keyword Spotting System Based on Large Vocabulary Continuous Speech Recognition and Linguistic Knowledge. *Journal of Electrical and Computer Engineering*, *2016*. 1-9.

[22] Kamarudin, M. R., Yusof, M. A. F. M., & Jaya, H. T. (2013). Low cost smart home automation via microsoft speech recognition. *International Journal of Engineering & Computer Science*, 13(3), 6-11.

[23] Chen, H. H. J. (2011). Developing and evaluating an oral skills training website supported by automatic speech recognition technology. *ReCALL*, *23*(1), 59-78.

[24] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Silovsky, J. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding* (No. EPFL-CONF-192584). IEEE Signal Processing Society.

[25] Tashev, I. (2013). Kinect development kit: A toolkit for gesture-and speech-based human-machine interaction [best of the web]. *IEEE Signal Processing Magazine*, *30*(5), 129-131.

[26] Maskeliunas, R., Ratkevicius, K., & Rudzionis, V. (2011). Voice-based human-machine interaction modeling for automated information services. *Elektronika ir Elektrotechnika*, *110*(4), 109-112.

[27] Krishna, Y. B., & Nagendram, S. (2012). Zigbee based voice control system for smart home. *International Journal on Computer Technology and Applications*, *3*(1), 163-168.

[28] Baig, F., Beg, S., & Khan, M. F. (2013). Zigbee based home appliances controlling through spoken commands using handheld devices. *International Journal of Smart Home*, *7*(1), 19-26.

[29] Sharma, F. R., & Wasson, S. G. (2012). Speech recognition and synthesis tool: assistive technology for physically disabled persons. *International Journal of Computer Science and Telecommunications*, *3*(4), 86-91.

[30] Yamamoto, K., Kassai, K., Kuramoto, I., & Tsujino, Y. (2017). Presenter Supporting System with Visual-Overlapped Positive Response on Audiences. In *Advances in Affective and Pleasurable Design* (pp. 87-93). Springer International Publishing.

[31] Rai, A., Khan, A., Bajaj, A., & Khurana, J. B. (2017). An efficient online examination system using speech recognition. *International Research Journal of Engineering and Technology*, 4(4), 2938-2941.

[32] Kumar, A. J., Schmidt, C., & Koehler, J. (2017). A Knowledge Graph Based Speech Interface for Question Answering Systems. *Speech Communication*, 92, 1-12.

[33] El Amrani, M. Y., Rahman, M. H., Wahiddin, M. R., & Shah, A. (2016). Towards using CMU Sphinx tools for the Holy Quran recitation verification. *Int. J. Islam. Appl. Comput. Sci. Technol*, *4*(2), 10-15.

[34] Phull, D. K., & Kumar, G. B. (2016). Investigation of indian english speech recognition using cmu sphinx. *International Journal of Applied Engineering Research*, *11*(6), 4167-4174.

[35] Satori, H., & ElHaoussi, F. (2014). Investigation Amazigh speech recognition using CMU tools. *International Journal of Speech Technology*, *17*(3), 235-243.

[36] Satori, H., Zealouk, O., Satori, K., & ElHaoussi, F. (2017). Voice comparison between smokers and non-smokers using HMM speech recognition system. *International Journal of Speech Technology*, 1-7.

[37] Kulkarni, K., Londhe, A., Mahajan, B., Inamdar, C., & Jakhotiya, A. (2016). Comprehensive Tool for Generation and Compatibility Management of Subtitles for English Language Videos. *International Journal of Computational Intelligence Research*, *12*(1), 63-68.

[38] Płonkowski, M., & Urbanovich, P. (2014). Tuning a CMU Sphinx-III speech recognition system for Polish language. *Przegląd Elektrotechniczny*, *90*(4), 181-184.

[39] Medennikov, I., & Prudnikov, A. (2016, August). Advances in STC Russian Spontaneous Speech Recognition System. In *International Conference on Speech and Computer* (pp. 116-123). Springer International Publishing.

[40] Besacier, L., Gauthier, E., Mangeot, M., Bretier, P., Bagshaw, P., Rosec, O., ... & Nocera, P. (2015, September). Speech Technologies for African Languages: Example of a Multilingual Calculator for Education. In *Interspeech 2015 (short demo paper)*, Dresden, Germany.

[41] Peddinti, V., Manohar, V., Wang, Y., Povey, D., & Khudanpur, S. (2016). Far-Field ASR Without Parallel Data. In *INTERSPEECH* (pp. 1996-2000), September 8–12, 2016, San Francisco, USA.

[42] Yamada, Y., Nose, T., Chiba, Y., Ito, A., & Shinozaki, T. (2017, August). Development and Evaluation of Julius-Compatible Interface for Kaldi ASR. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (pp. 91-96). Springer, Cham.

[43] Sakai, K., Ishi, C. T., Minato, T., & Ishiguro, H. (2015, August). Online speech-driven head motion generating system and evaluation on a tele-operated robot. In *Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on* (pp. 529-534). IEEE.

[44] Lojka, M., Ondas, S., Pleva, M., & Juhar, J. (2014). Multi-thread parallel speech recognition for mobile applications. *Journal of Electrical and Electronics Engineering*, *7*(1), 81-86.

[45] Galić, J., Jovičić, S. T., Grozdić, Đ., & Marković, B. (2014, October). HTK-based recognition of whispered speech. In *International Conference on Speech and Computer* (pp. 251-258). Springer, Cham.

[46] Mankala, S. R., Bojja, S. R., Ramaiah, V. S., & Rao, R. R. (2014). Automatic speech processing using HTK for Telugu language. *International Journal of Advances in Engineering & Technology*, *6*(6), 2572-2578.
[47] Adetunmbi, O. A., Obe, O. O., & Iyanda, J. N. (2016). Development of Standard Yorùbá speech-to-text system using HTK. *International Journal of Speech Technology*, *19*(4), 929-944.
[48] Bougares, F., Deléglise, P., Esteve, Y., & Rouvier, M. (2013, September). LIUM ASR system for Etape French evaluation campaign: experiments on system combination using open-source recognizers. In *International Conference on Text, Speech and Dialogue* (pp. 319-326). Springer, Berlin, Heidelberg.
[49] Akram, H., & Khalid, S. (2016). Using features of local densities, statistics and HMM toolkit (HTK) for offline Arabic handwriting text recognition. *Journal of Electrical Systems and Information Technology*, 3(3), 99-110.

Rami Matarneh. "Speech Recognition Systems: A Comparative Review." IOSR Journal of Computer Engineering (IOSR-JCE) , vol. 19, no. 5, 2017, pp. 71–79.