SYNCHROINFO JOURNAL

www.media-publisher.eu/synchroinfo-journal

SPEECH SIGNAL PROCESSING METHODS IMPLEMENTATION IN WEB APPLICATION DEVELOPMENT

M. V. Galickiy [1]

[1] Network Information Technologies and Services, MTUCI, Moscow, Russia

m.v.galickiy@mtuci.ru

ABSTRACT

Information systems are becoming more complex due to the integration of artificial intelligence and machine learning. The introduction of additional data entry methods has the potential to increase user productivity. To improve the accuracy and efficiency of speech-to-text conversion, it is essential to consider technologies such as voice activity detection and automatic speech recognition. They provide advanced mechanisms for user-system interaction through natural user interfaces, in particular, voice. The article will also discuss some ASR platforms with different levels of adaptation to linguistic and acoustic environments. The subject of research in this article is the methods of voice activity detection (VAD) and automatic speech recognition (ASR). The purpose of the study is to analyze the VAD and ASR modules and test them to make recommendations on their use. The results of the study will be useful for web application developers who are thinking about implementing this modules in their projects.

KEYWORDS: *voice recognition; algorithms; web application; speech synthesis; ASR; VAD; artificial intelligence*

DOI: 10.36724/2664-066X-2025-11-4-2-8

Received: 20.06.2025
Accepted: 23.08.2025

Citation: M. V. Galickiy, "Speech signal processing methods implementation in web application development", *Synchroinfo Journal* **2025**, vol. 11, no. 4, pp. 2-8.

Introduction

Currently, many organizations operate in a rapidly changing environment, where information systems (IS) are becoming more complex due to the integration of artificial intelligence and machine learning. In most computer workstations, manual input remains the main method for entering information. However, the introduction of additional data entry methods has the potential to increase user productivity.

To improve the accuracy and efficiency of speech-to-text conversion, it is essential to consider technologies such as voice activity detection (VAD) and automatic speech recognition (ASR). They provide advanced mechanisms for user-system interaction through natural user interfaces (NUI), in particular, voice. The article will also discuss some ASR platforms with different levels of adaptation to linguistic and acoustic environments.

Overview of ASR platforms and their capabilities

Mozilla DeepSpeech [1], CMU Sphinx [2], Vosk [3] and the Google Speech API are all interfaces that support speech recognition and synthesis for web applications. They allow you to create programs that can listen to the user and respond to them with a voice. These interfaces differ in their architecture, principles of operation, and scope of application. However, they share two key features:

1. A component that enables user speech recognition. This component enables listening to audio data from a microphone, converting it into text information, and creating convenient and intuitive user interfaces based on voice commands.
2. The component that enables the synthesis of speech and conversion of text information into audio data, thus allowing the program to produce a voice response.

By using these ASR platforms, developers have a wide range of options for working with voice technologies on the web platform. For example, you can create web applications that use built-in tools to recognize user speech. This feature is useful for creating voice-based searches, voice-controlled interfaces, and other applications where it is more convenient to interact with speech rather than text [4].

In addition, they are an excellent tool for creating voice assistants. With the help of this technology, applications can access a device's microphone, listen to audio, and transmit it for processing. They can then respond to the user's voice commands.

As for speech synthesis, the second feature allows you to convert text information into spoken words, making interfaces more accessible to users with hearing impairments.

However, it is important to note that these platforms do have some limitations. For instance, the Google Speech API cannot be used offline and requires a continuous Internet connection. Mozilla DeepSpeech needs a significant amount of resources for training, while Vosk consumes a significant amount of RAM. These factors should be taken into consideration when testing and using these tools.

Voice activity detection and automatic speech recognition

Automatic speech recognition (ASR) technologies have now reached a level of sophistication that they can be reliably used to improve web user interfaces. Some of the most important user operations include:

1. website navigation, including clicking on links that are not provided on the current page;
2. filling out input form (e.g., text fields, number fields, drop-down lists);
3. performi actions (for example, submitting or canceling completed forms).

Automatic speech recognition (ASR) and analysis of the algorithms used

Automatic speech recognition (ASR) technology allows electronic devices to recognize spoken words and has been the subject of research since the 1950s [5]. ASR can be seen as a mathematical model that transforms speech audio signals into text. It's important to differentiate ASR from voice biometrics, which is focused on identifying the speaker rather than the speech content [6].

Automatic speech recognition (ASR), now implemented in voice assistants, is an additional input method for devices such as mobile phones, tablets, and virtual assistants. Taking advantage of the widespread demand, ASR technologies have reached a level of maturity that justifies their use in web systems as an additional method of information input.

Among the approaches to automatic speech recognition (ASR), hidden Markov models (HMM) and deep neural networks (DNN) have been widely studied [7].

HMM is a statistical model developed in the 1960s that describes sequences of events and the probabilities of transitions between them. This algorithm has found wide application in speech recognition, as it allows modeling various phonetic units and predicting the most probable word order [8].

The HMM is a relatively simple and effective algorithm that can be used to recognize various types of sounds, including speech. It is based on the assumption that each observed position is the result of the previous position, with a certain transition probability. A device using this algorithm analyzes the data heard and predicts what word or sound will come next.

In order for the device to function with HMM, a set of training data is required. This includes sequences of observed symbols and corresponding sequences of hidden states. These parameters, such as the probability of transitions between states and the probability of observed symbols for each state, are used to train the HMM using mathematical techniques, such as the maximum likelihood method or the Baum-Welsh algorithm [9].

Once trained, the HMM can be used for speech recognition. This requires obtaining a sequence of observed symbols corresponding to the audio signal and using the model to determine the most probable sequence of hidden states corresponding to this data.

The advantages of this algorithm include its high speed of sound and speech recognition, even in noisy or distorted environments. Additionally, it has the benefit of having access to training data, allowing you to tailor the model to specific tasks. However, this data needs to be presented in large quantities in order to ensure high accuracy in recognition. The disadvantages also include the need for manual adjustment of model parameters and the potential for inaccurate results when detecting complex sounds.

Another algorithm is Deep Neural Networks (DNN). DNN is one of the most popular methods in the field of speech recognition. It consists of multi-layered networks of artificial neurons that are trained on a large amount of labeled data in order to achieve high accuracy in recognition [10].

Each layer of DNN neurons performs a specific function. The first layer of the neural network accepts audio data as input. This data is then processed by subsequent layers, which produce the final output. The number of layers and neurons in each layer is determined by the network's architecture and the specific tasks it is designed to solve.

DNN training involves adjusting the weights and biases of each neuron in the network to allow it to correctly process audio data. This is done using the backpropagation method, which allows the importance of different parts of the data to be adjusted based on the difference between actual and expected results.

DNN, like HMM, can achieve high speech recognition accuracy if high-quality training data is available in large quantities. But unlike HMM, DNN is self-learning, which allows it to be used to solve problems without explicit class labels. The disadvantage of this algorithm is the requirement for large computing resources and the choice of network architecture, which makes it difficult to use on mobile devices and other limited systems [11].

In most modern ASR systems, the audio signal is converted into a set of vector features, which are then used in subsequent processing stages. This process is sensitive to noise, accent, age, and gender of the speaker. In addition, context-independent and language models are used. The former is trained to recognize phonemes from a feature vector, which is used to construct words. The language model is responsible for grammatical rules and determines the most likely word order in a sentence. It is usually represented by n-gram models containing statistical data on word sequences.

Currently, many speech recognition systems are used as virtual assistants on mobile devices. For example, the most popular voice assistants based on this system are Alice from Yandex, Marusya from VK and Salute from Sber.

There are many commercial ASR platforms offering ready-made integration solutions, including the Microsoft Bing Speech API, Google Speech API, and IBM Watson Speech-to-Text.These platforms are available under a license. However, there are open source toolkits that allow developers to create their own speech-to-text conversion systems for various programming languages and platforms, such as Mozilla DeepSpeech, CMU Sphinx, and Vosk, already mentioned above. Google Speech API is also used for comparative analysis, as it has a high accuracy in speech recognition.

Mozilla DeepSpeech uses a recurrent neural network (RNN) architecture, which is implemented using the TensorFlow framework.

CMU Sphinx is a popular platform among the scientific community, offering a wide range of tools and a flexible design. This allows for the quick and easy development of speech recognition (ASR) applications.

Vosk API is a speech recognition tool that integrates offline models for 17 languages.

Voice activity detection (VAD)

Voice Activity Detection (VAD) refers to signal processing techniques used to detect speech in an audio signal. In speech processing systems, the problem of distinguishing between speech and non-speech signals remains relevant, especially for web applications operating in real time. Speech processing algorithms often place high demands on computational resources. However, speech is inherently intermittent, and incorporating VAD into these algorithms is an optimization strategy to reduce unnecessary computation [12].

VAD methods differ in their processing principles, but their main goal is to extract speech data from a given audio signal and separate it from non-speech fragments. In most cases, speech fragments are grouped for further processing, which allows noise to be removed from the input data.

In general, VAD methods can be classified into two categories:

1. Energy threshold-based methods. These methods rely on the fact that speech adds energy to the signal. This method allows one to distinguish between high- and low-energy regions, i.e., regions without speech. This approach is simple to implement and is widely used in systems with limited computing resources.

2. Machine learning-based methods. These methods involve selecting one or more speech characteristics, using learning algorithms, and training on large amounts of data suitable for the intended use cases. Despite their high accuracy, such methods require significant computational resources, and their implementation remains complex and requires further improvement [13].

Module results VAD

Any technology must meet several requirements in order to have the desired impact on users. In this case, the processing time from the moment the audio signal is captured to the execution of the associated operation is a critical factor in ensuring a smooth workflow. Excessive processing time may negatively impact system acceptance and implementation.

In addition, there are two major issues that may be of concern to both end users and company management: data privacy and information security [14-18]. However, these issues are beyond the scope of this study.

For voice-related applications, it is important to save bandwidth by disabling streaming when no voice content is detected. In speech recognition tasks, the processor load can be reduced by avoiding processing unnecessary fragments without speech content. On the other hand, if the speech signal is incorrectly classified as noise, the module will not be able to transmit all the necessary information for further recognition [19].

For testing, a VAD module was used, implemented based on the energy threshold level, having a binary output, where audio signals are classified as speech or non-speech.

To evaluate the performance of the VAD module, a criterion related to the correct segmentation of speech content was adopted. First of all, no important speech content that needs to be processed should be blocked. The Speech Hit Rate (SHR) [20], which reflects the percentage of correct speech detection, was used as an objective evaluation parameter. Additionally, the Noise Suppression Ratio (NSR) was used, which measures the amount of noise that was blocked relative to its total volume in the sound segment.

Five SNR (signal-to-noise ratio) levels were used in the testing:

• cl an sound,

• SNR = 0, 5, 10, and 15.

Three audio files with speech superimposed on background crowd noise (Sp01, Sp02, and Sp03) were tested. Each test audio segment at a given SNR level included three speech samples separated by noise fragments.

Figure 1 demonstrates the results of the VAD module for SNR = 10 (the dashed blue vertical lines indicate spch ON and the red ones indicate spch OFF, the real parts of the speech data are marked manually with a green background), and Table 1 shows the summary test results for each SNR level (the results of the speech hit rate (SHR) and noise reduction ratio (NSR) assessment).

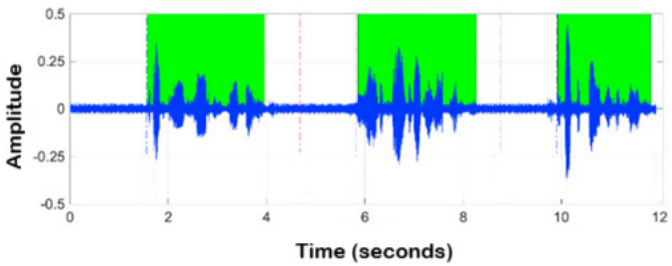

Figure 1. Results of the VAD module

Table 1

Summary test results

SNR	SNR sp01	SNR sp02	SNR sp03	NSR
clean	0.98	0.98	0.98	0.81
15	0.98	0.98	0.98	0.70
10	0.98	0.98	0.98	0.71
5	0.97	0.99	0.99	0.80
0	1.0	1.0	1.0	0.09

Module results ASR

Mozilla DeepSpeech, CMU Sphinx, Vosk and Google Speech API were used for testing.

DeepSpeech training involved data augmentation applied to 15% of the original dataset. The main hyperparameters used during training included:

• dr p_source_layers = 5 (to adjust all model weights),

• batch s ze = 16 (to account for hardware capabilities),

• num er of epochs = 200,

• n_h dden = 2048 (as recommended for DeepSpeech),

• learning_rate = 0.0001, and dropout_rate = 0.05, which yielded acceptable results after trial and error.

CMU Sphinx training was performed using sphinxtrain4 compiled for Linux. The training steps included:

1. creat g a dictionary with the required vocabulary,
2. setting u a phoneme file,
3. creat g a language model using the CMU Sphinx online tool.

In addition, a five-fold cross-validation process was conducted, where 80% of the data was used for training and 20% for validation.

To compare Vosk, DeepSpeech, CMU Sphinx, and the Google Speech API, we used a test dataset belonging to the best model obtained during cross-validation for CMU Sphinx.

For each audio file, the following were calculated:

• LD (Levenshtein Distance) – a measure of recognition errors;

• processing time (inference time).

The results are presented as histograms for English and Russian languages (Fig. 2 and Fig. 3).

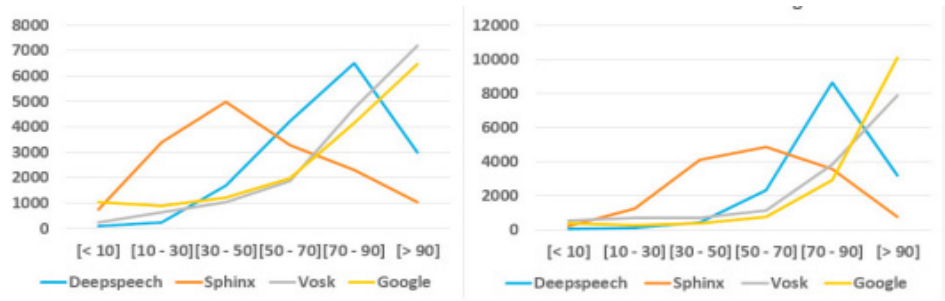

Figure 2. Results (Levenshtein Histogram)

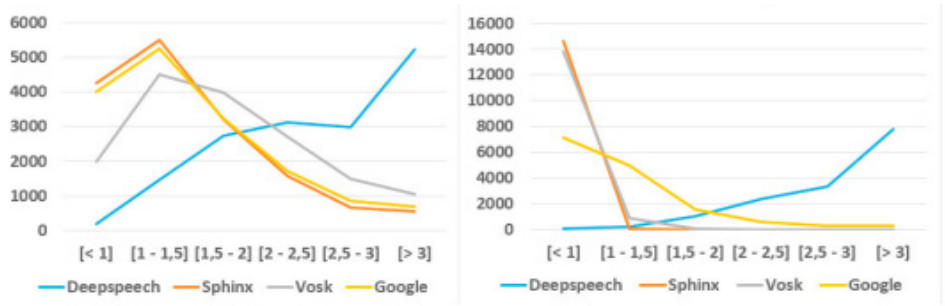

Figure 3. Results (Time Histogram)

Conclusion

In this article, the methods of voice activity detection (VAD) and automatic speech recognition (ASR) were reviewed and tested, and their effectiveness and accuracy were evaluated. The following results were obtained in the course of the study:

The VAD algorithm, implemented based on an energy threshold, does not guarantee effective speech-noise discrimination under conditions where the SNR is below 5.

At SNR = 0, an extreme case occurs: the algorithm is unable to separate speech from noise and classifies the entire signal as speech data.

For ASR:

• Vosk shows similar results t the commercial Google Speech API;

• Vosk outperforms other systems in recognition accuracy for two languages, achieving an accuracy of over 85%;

• D epSpeech ranked second in accuracy (according to the LD metric);

• C U Sphinx was the fastest in processing time (less than 1 second);

• Vosk showed a processing time of 1.0–1.5 seconds, similar to the Google Speech API;

• DeepSpeech was the slowest, with an average processing time of over 2.5 seconds for English and over 3 seconds for Russian.

REFERENCES

[1] DeepSpeech's documentation [Electronic resource]. Mode of access: https://deepspeech.readthedocs.io/en/latest/ (Date of access: 07.07.2025)

[2] Cmusphinx [Electronic resource]. Mode of access: https://cmusphinx.github.io/wiki/about/ (Date of access: 07.11.2025)

[3] Vosk Offline speech recognition API [Electronic resource]. Mode of access: https://alphacephei.com/vosk/ (Date of access: 07.07.2025)

[4] ASR [Electronic resource]. Mode of access: https://sonix.ai/resources/what-asr/ (Date of access: 07.07.2025)

[5] S. Furui, "Speech Recognition – Past, Present, and Future," *NTT review*, vol. 7, no. 2, 1995, pp. 13-18.

[6] R.S. Rocha, P. Ferreira, I. Dutra, R. Correia, R. Salvini, E. Burnside, "A Speech-to-Text Interface for MammoClass," *2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)*, 2016, pp. 1-6.

[7] M. Bohac, M. Kucharova, Z. Callejas, J. Nouza, P. Červa, "A cross-lingual adaptation approach for rapid development of speech recognizers for learning disabled users," *EURASIP Journal on Audio, Speech, and Music Processing*, 2014.

[8] P. Barry, P. Crowley, "Modern Embedded Computing: Designing Connected, Pervasive, Media-Rich Systems," 2012, pp. 16-19.

[9] Hidden Markov chains [Electronic resource]. Mode of access: https://habr.com/ru/articles/188244/ (Date of access: 07.07.2025).

[10] DNN Neural Network [Electronic resource]. Mode of access: https://www.educba.com/dnn-neural-network/ (Date of access: 07.07.2025).

[11] B. Lindberg, "Low-Complexity Variable Frame Rate Analysis for Speech Recognition and Voice Activity Detection," *IEEE Journal of Selected Topics in Signal Processing*, 2010, pp. 798-807.

[12] A Real-Time Voice Activity Detection Algorithm [Electronic resource]. Mode of access: https://www.pvsm.ru/programmirovanie/42828 (Date of access: 07.07.2025).

[13] Skillbox media [Electronic resource]. Mode of access: https://skillbox.ru/media/code/kak-ustroeno-mashinnoe-obuchenie-zadachi-algoritmy-i-vidy-machine-learning/ (Date of access: 07.07.2025).

[14] V. A. Dokuchaev, "The impact of new information and communication technologies on the privacy of personal data," *Current problems and prospects of economic development: XXIII International Scientific and Practical Conference*, 2024, pp. 12-15.

[15] V. A. Dokuchaev, V.V. Maklachkova, A. A. Boiko, "The problem of data updating in CRM systems," *Economics and quality of communication systems*, 2025, no. 1(35), pp. 45-57.

[16] V.Y. Statev, V. A. Dokuchaev, V.V. Maklachkova, "Information security in the big data space," *T-Comm*. 2022. Vol. 16, no. 4, pp. 21-28. DOI 10.36724/2072-8735-2022-16-4-21-28.

[17] V. A. Dokuchaev, V. V. Maklachkova, V. Yu. Statev, "Classification of personal data security threats in information systems," *T-Comm*. 2020. Vol. 14, no. 1, pp. 56-60. DOI 10.36724/2072-8735-2020-14-1-56-60.

[18] V. A. Dokuchaev, "Digital transformation: New drivers and new risks," *2020 International Conference on Engineering Management of Communication and Technology, EMCTECH 2020* : Proceedings, Vienna, 2020. New York: Institute of Electrical and Electronics Engineers Inc., 2020. P. 9261544. DOI 10.1109/EMCTECH49634.2020.9261544.

[19] How ASR works [Electronic resource]. Mode of access: https://cloud.vk.com/blog/slushaet-i-ponimaet-kak-rabotaet-tehnologija-avtomaticheskogo-raspoznavanija-rechi/ (Date of access: 07.07.2025) (in Russian).

[20] Efficient voice activity detection algorithm [Electronic resource]. Mode of access: https://asmp-eurasipjournals.springeropen.com/articles/10.1186/1687-4722-2013-21 (Date of access: 07.07.2025).