



Automatic Speech Recognition Using Limited Vocabulary: A Survey

Jean Louis K. E Fendji, Diane C. M. Tala, Blaise O. Yenke & Marcellin Atemkeng

To cite this article: Jean Louis K. E Fendji, Diane C. M. Tala, Blaise O. Yenke & Marcellin Atemkeng (2022) Automatic Speech Recognition Using Limited Vocabulary: A Survey, Applied Artificial Intelligence, 36:1, 2095039, DOI: [10.1080/08839514.2022.2095039](https://doi.org/10.1080/08839514.2022.2095039)

To link to this article: <https://doi.org/10.1080/08839514.2022.2095039>



© 2022 The Author(s). Published with license by Taylor & Francis Group, LLC.



Published online: 25 Jul 2022.



Submit your article to this journal [↗](#)



Article views: 3036



View related articles [↗](#)



View Crossmark data [↗](#)






Citing articles: 6 View citing articles [↗](#)

REVIEW

 OPEN ACCESS  Check for updates

Automatic Speech Recognition Using Limited Vocabulary: A Survey

Jean Louis K. E Fendji ^a, Diane C. M. Tala^b, Blaise O. Yenke ^a,
and Marcellin Atemkeng ^c

^aDepartment of Computer Engineering, University Institute of Technology, University of Ngaoundere, Cameroon; ^bDepartment Mathematics and Computer Science, Faculty of Science, University of Ngaoundere, Cameroon; ^cDepartment of Mathematics, Rhodes University, Grahamstown South Africa

ABSTRACT

Automatic Speech Recognition (ASR) is an active field of research due to its large number of applications and the proliferation of interfaces or computing devices that can support speech processing. However, the bulk of applications are based on well-resourced languages that overshadow under-resourced ones. Yet, ASR represents an undeniable means to promote such languages, especially when designing human-to-human or human-to-machine systems involving illiterate people. An approach to design an ASR system targeting under-resourced languages is to start with a limited vocabulary. ASR using a limited vocabulary is a subset of the speech recognition problem that focuses on the recognition of a small number of words or sentences. This paper aims to provide a comprehensive view of mechanisms behind ASR systems as well as techniques, tools, projects, recent contributions, and possible future directions in ASR using a limited vocabulary. This work consequently provides a way forward when designing an ASR system using limited vocabulary. Although an emphasis is put on limited vocabulary, most of the tools and techniques reported in this survey can be applied to ASR systems in general.

ARTICLE HISTORY

Received 28 April 2022
Accepted 21 June 2022

Abbreviations

ACC: Accuracy; AM: Acoustic Model; ASR: Automatic Speech Recognition; BD-4SK-ASR: Basic Dataset for Sorani Kurdish Automatic Speech Recognition; CER: Character Error Rate; CMU: Carnegie Mellon University; CNN: Convolutional Neural Network; CNTK: CogNitive ToolKit; CUED: Cambridge University Engineering Department; DCT: Discrete Cosine Transformation; DL: Deep Learning; DNN: Deep Neural Network; DRL: Deep Reinforcement Learning; DWT: Discrete Wavelet Transform; FFT: Fast Fourier Transformation; GMM: Gaussian Mixture Model; HMM: Hidden Markov Model; HTK: Hidden Markov Model ToolKit; JASPER: Just Another Speech Recognizer; LDA: Linear Discriminant Analysis; LER: Letter Error Rate; LGB: Light Gradient Boosting Machine; LM: Language Model; LPC: Linear Predictive Coding; LVCSR: Large Vocabulary Continuous Speech Recognition; LVQ: Learning

CONTACT Jean Louis K. E Fendji  lfendji@gmail.com  Department of Computer Engineering, University Institute of Technology, University of Ngaoundere, 455 Ngaoundere, Cameroon; Marcellin Atemkeng  m.atemkeng@ru.ac.za  Department of Mathematics, Rhodes University, Grahamstown 6140, South Africa

© 2022 The Author(s). Published with license by Taylor & Francis Group, LLC.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Vector Quantization Algorithm; MFCC: Mel-Frequency Cepstrum Coefficient; ML: Machine Learning; PCM:Pulse-Code Modulation; PPVT: Peabody Picture Vocabulary Test; RASTA: RelAtive SpecTral; RLAT: Rapid Language Adaptation Toolkit; S2ST: Speech-to-Speech Translation; SAPI: Speech Application Programming Interface; SDK: Software Development Kit; SVASR:Small Vocabulary Automatic Speech Recognition; WER: Word Error Rate

Introduction

Automatic speech recognition (ASR) is the process and the related technology applied to convert a speech signal into the matching sequence of words or other linguistic entities using algorithms implemented in computing devices (Indurkha and Damerau 2010). ASR has become an exciting field for many researchers. Presently, users prefer to use devices such as computers, smartphones, or any other connected device through speech. Current speech processing techniques (encompassing speech synthesis, speech processing, speaker identification or verification) pave the way to create human-to-machine voice interfaces. ASR can be applied in several applications including voice services (Yadava and Jayanna 2017), program control and data entry (Hauser, Sabir, and Thoma 1999), avionics (Noyes and Starr 2007), disabled assistance (Mayer 2018), amongst others. Although ASR can be advantageous in easing human-to-machine communication; in many cases, it goes beyond helpful and becomes absolutely necessary. For example, low-literacy levels and the extinction of under-resourced languages are ideal candidates for ASR. (Besacier et al. 2014). In fact, the high penetration of communication tools such as smartphones in the developing world (Albaptain et al. 2014) and their increasing presence in rural areas (Ebongue 2015; Ebongue Louis 2015) provides an unprecedented opportunity to develop a voice-based application that can help to mitigate the low literacy levels in those areas. Smartphones offer many advantages over a PC-based interface, such as high mobility and portability, easy recharge of their batteries, and conventional embedded features such as microphones and speakers.

Motivation

In regions with low literacy levels, people are used to speaking local languages that are often considered as under-resourced languages because of the lack or insufficiency of formal written grammar and vocabulary. Since people do not know how to read or to write well-resourced languages (such as English or French), the development of ASR systems for under-resourced languages appears as an appealing solution to overcome this limitation. However, due to the complexity of the task, limited vocabulary must be considered.

This paper focuses on limited vocabulary in ASR to allow researchers who wish to work on under-resourced languages to have an overview on how to develop a speech recognition system for limited vocabulary. In contrast to limited vocabulary systems, large vocabulary continuous speech recognition (LVCSR) systems are usually trained on thousands of hours of speech and billions of words of text (Saon and Chien 2012). The development of large vocabulary systems is complex since the larger the vocabulary, the harder the manipulation of learning algorithms, with more rules needed to build the dataset. LVCSR systems can be very efficient when they are applied on similar domains to those on which they were trained (Orosanu and Jovet 2018). However, they are not robust enough to handle mismatched training and test conditions as the context may not be well handled. In fact, most of the input can be silence or contain background noise, which can be mistaken for speech; this increases the false positive rate (Warden 2018). Thus, LVCSR systems are not suitable for transfer learning targeting small or limited vocabulary.

Position with Other Surveys

Extensive research has been done regarding speech recognition using limited vocabulary. Among the existing surveys, authors in (Lima and Costa-Abreu 2020) focus on Portuguese-based language (and variations). The authors consider Portuguese as an understudied language compared to English, Arabic, and Asian languages. Among Asian languages, Indian languages received particular attention. Works on the development of ASR systems dealing with Indian languages such as Hindi, Punjabi, Tamil, amongst other, are presented in (Kurian 2014). Another specific language survey is provided in (Ronzhin et al. 2006), where the authors focus on Russian language specificities and apply models for the development of Russian speech recognition systems in some organizations, both in Russia and abroad. A broader survey inspecting more than 120 promising works on biometric recognition (including voice) based on deep learning (DL) models is provided in (Minaee et al. 2020). In the latter, the authors present the strengths and possible uses of DL. A narrowed survey is presented in (Desai, Dhameliya, and Desai 2013; Mishaim et al. 2021) and highlights the major subjects and improvements made in ASR. Additionally, a conical survey focusing on various feature extraction techniques in speech processing is provided in (Hibare and Vibhute 2014). The work in (Gong 1995) attempts to provide a comprehensive survey on noise-resistant features as well as similarity measurement, speech enhancement, and speech model compensation in noisy contexts. Due to the increasing penetration rate of mobile devices, the work in (Zaykovskiy 2006) investigates different approaches for providing ASR technology to mobile users. Approaches used to design Chatbots and a comparison between different design techniques from nine papers are given in (Abdul- Kader & Woods, 2015). To evaluate the

reliability of recognition results, the work in (Jiang 2005) summarizes most research works related to confidence measures. One of the first comprehensive surveys on speech recognition systems for under-resourced languages is found in (Besacier et al. 2014), notwithstanding the fact that many of the issues and approaches presented in the paper apply to speech technology in general. In this survey, authors do not focus on limited vocabulary. The authors in (Rajashri, Ambewadikar, and Baheti 2021) have done a review for ASR for small vocabulary, but in this survey they do not clearly describe the methods and techniques used to build the models. Moreover this survey is limited to the Marathi language only. Table 1 provides a summary of the recent surveys on ASR.

Contributions

Despite the plethora of surveys, a survey on ASR using limited vocabularies is yet to be conducted. However, ASR using limited vocabulary is a tremendous opportunity as a starting point for the development of speech recognition systems for under-resourced languages. This survey helps to fill this gap by presenting a summary of works done on ASR for limited vocabulary. For a better understanding, the ASR principle is detailed along with the approach to i) build ASR systems, ii) construct datasets and iii) evaluate the performance of such systems. Furthermore, close and open-source toolkits, and frameworks are also presented. Therefore, such a study can rapidly and easily enable researchers who want to build speech recognition systems using limited vocabulary. The contributions of this paper are as follows:

- A description of fundamental aspects of ASR;
- A description of tools and processes for creating ASR systems;
- A summary of important contributions in ASR with limited vocabulary; and
- Orientations for future works.

The rest of the paper is organized around eleven sections. Section 2 presents the methodology used to conduct this survey. Section 3 provides an understanding of “limited vocabulary,” and Section 4 describes the principle of speech recognition. Section 5 describes the techniques used for ASR, and Section 6 deals with the management of datasets. Section 7 presents the traditional performance metrics, and Section 8 provides an insight into the speech recognition frameworks. Section 9 summarizes works on speech recognition using limited vocabulary. Section 10 discusses possible future directions, followed by a conclusion in Section 11.

Table 1. Summary of the recent surveys on ASR with objective and limitation.

Reference Year	Objective of the review	Limitation	Language specific	Limited vocabulary
(Gong 1995)	To present noise resistant features and similarity measurement, speech enhancement and speech model compensation in noisy environments	No description of the construction mechanism of an ASR.	No	No
(Ronzhin et al. 2006)	To expose specificities, methods, and applied models for the development of Russian speech recognition.	Although the authors provided a broad view of works related to ASR for Russian, they did not describe the ASR development methods. They focused on systems and their performance.	Russian	No
(Zaykovskiy 2006)	To detail approaches for providing ASR technology to mobile users	Only interested in ASR architecture. No insight on dataset or corpora.	No	No
(Desai, Dhameiya, and Desai 2013)	To summarize some of the well-known methods used in several stages of ASR systems.	Provide insight on speech recognition techniques, feature extraction techniques and traditional performance metrics.	No	No
(Kurian 2014)	To present major research works in development of ASR in Indian languages.	Focuses on Indian languages and does not describe either ASR development methods or dataset construction.	Indian	No
(Hibare and Vibhute 2014)	To present various feature extraction techniques in speech processing.	Focuses solely on extraction techniques.	No	No
(Besacier et al. 2014)	To discuss speech recognition system for under-resourced languages.	Provides a good understanding of under-resourced languages and mechanism for ASR, but does not describe dataset design mechanism.	No	No
(Abdul-Kader and Woods 2015)	To present techniques used to design chatbots.	Does not detail the speech recognition mechanism or the procedure for creating dataset.	No	No
(Lima & Costa- Abreu, 2020)	To present Portuguese-based ASR.	Focuses only on Portuguese-based ASR (corpora, approaches), even if tools can be used regardless of the language.	Portuguese	No
(Minaee et al. 2020)	To provide an insight of Biometric recognition including voice and based on DL. models Not specific to speech recognition.	Only provides insight with no description of the ASR mechanism or dataset construction.	No	No
(Rajashri, Ambewadikar, and Baheti 2021)	To review ASR system or interfaces for specific tasks.	Limited to ASR for small vocabulary for Marathi language only.	Marathi	Yes
(Mishaim et al. 2021)	To present a comparison between ASR techniques nowadays used and deep learning methods .	The authors do not dwell on the dataset construction techniques used in the different papers cited.	No	No

Methodology

The research methodology is composed of three main steps, namely the collection of papers, filtering to keep only relevant papers with significant findings, and analysis of the selected papers. The procedure used for collection and filtering are detailed below.

Collection

The collection of papers has been performed by a keyword-based search from common sources, such as Scopus, IEEE Xplore, ACM, ScienceDirect, PubMed. In addition, the search has been extended to web scientific indexing services, namely Web of Science and Google Scholar. The aim was to collect as many relevant papers as possible regarding ASR. For that purpose, several keywords have been used, such as: “Speech Recognition”] AND [“limited vocabulary” OR “vocabulary” OR “commands”].

Filtering

This step consists of identifying relevant papers by reading the abstract and screening the papers. Papers that do not directly deal with the topic or provide substantial contributions have been removed. However, some of papers providing relevant results have not been considered for the synthesis because they did not provide most of the needed parameters, such as the speech recognition language, the error rate, the type of environment, the exact size of the vocabulary and the number of speakers. We have selected 30 papers published between 2000 and 2021 for data synthesis.

Analyzing

Each paper has been analyzed based on the following criteria: the language for which the recognition system was designed; the toolkit used to design the models; the size of the dataset; the type of environment (noisy or not) for the recognition; the number of speakers considered for the elaboration of the dataset; and the accuracy or recognition rate.

Understanding of Limited Vocabulary

A vocabulary, in our context, is defined as a closed list of lexical units that can be recognized by an ASR system. The size of the vocabulary and the selection of lexical units in the vocabulary strongly influence the performance of the automatic transcription system since not all words outside the vocabulary can be recognized by the system. Speech recognition systems

basically access a dictionary of phonemes and words. Obviously, it is easier to seek the meaning of one out of ten words in a ten-word dictionary rather than one out of thousands of words in a Webster's dictionary (Agnes and Guralnik 1999).

A phoneme is, in essence, the smallest unit of phonetic speech that distinguishes one word from another (Clements 1985). Every word can be deconstructed into units (phonemes) of individual sounds that constitutes that word. The number of words or sentences an ASR system can recognize is an important classification criterion. As proposed in (Saksamudre, Shrishrimal, and Deshmukh 2015), an ASR system can be classified based on the size of the vocabulary as follows: Small Vocabulary (up to 100 words or sentences); Medium Vocabulary (between 101 and 1000 words or sentences); Large Vocabulary (between 1001 and 10,000 words or sentences); and Very-large vocabulary (more than 10,000 words or sentences).

A second classification is proposed by Whittaker and Woodland (Whittaker and Woodland 2001). They also classified ASR systems into four categories: small vocabulary (up to 1000 words), medium vocabulary (up to 10,000 words), large vocabulary (up to 100,000 words) and very/extra large vocabulary (up to 100,000 words). In this second case, limited vocabulary can be considered as a small vocabulary meaning up to 1000 words. This size will be used in the rest of the paper as the maximum size for limited vocabulary.

Automatic Speech Recognition

In the basic principle of ASR, the person speaking emits pressure variations in his larynx. The sounds produced are digitized by the microphone and transmitted through a medium or a network. Digitized sounds are transformed into acoustic units (or acoustic vectors) via an acoustic model (AM). Thereafter, the recognition engine analyses this sequence of acoustic vectors by comparing it with those in its memory (its language model [LM]) and proposes the most likely candidate sequence. It is therefore necessary for the sequence of acoustic vectors to approximate one of the sequences memorized by the recognition engine. The core of an ASR system can be seen as a mathematical model that can generate a text corresponding to the recognized pieces of speech input (Ghai and Singh 2012).

Architecture

Audio signals need to be digitized before the recognition process starts. The digitization of the signal requires the selection of an appropriate sampling frequency to catch the high-pitched voices (Kraleva and Kraleev 2009). In general, all ASR systems have the same architecture whether the vocabulary

is limited, medium, large, or very large. This architecture can be modified or supplemented according to the recognition to be performed. ASR is usually composed of five typical components: Feature extraction; Acoustic Model; Language Model; Pronunciation Model; and Decoder.

In the architecture illustrated in [Figure 1](#), the speech signal is received and then features are extracted. The obtained parameters are passed to the decoder, which uses the language, the acoustic, and the pronunciation models for learning.

Feature Extraction

Feature extraction is the first step for an ASR system. It converts the waveform speech signal to a set of feature vectors with the aim of having high discrimination between phonemes (Lokhande 2015). The feature extraction performs all the required measurements on the selected segment that will be used to make a decision (Doukas, Bardis, and Markovskiyi 2017). The measured features may be used to update long-term statistical measures to facilitate the adaptation of the process to varying environmental conditions (mainly the background) (Doukas and Bardis 2017). Feature extraction will determine the voice areas in the recording to be written out and extract sequences of acoustic parameters from them. There are many techniques for feature extraction, as reported in (Narang and Gupta 2015), including:

- Linear Predictive coding (LPC) (O'Shaughnessy 1988): LPC is a technique for signal source modeling in speech signal processing.
- RelAtive SpecTral (RASTA) filtering (Hermansky and Morgan 1994): RASTA is designed to decrease the impact of noise as well as heighten speech. This technique is widely used for noisy speech.

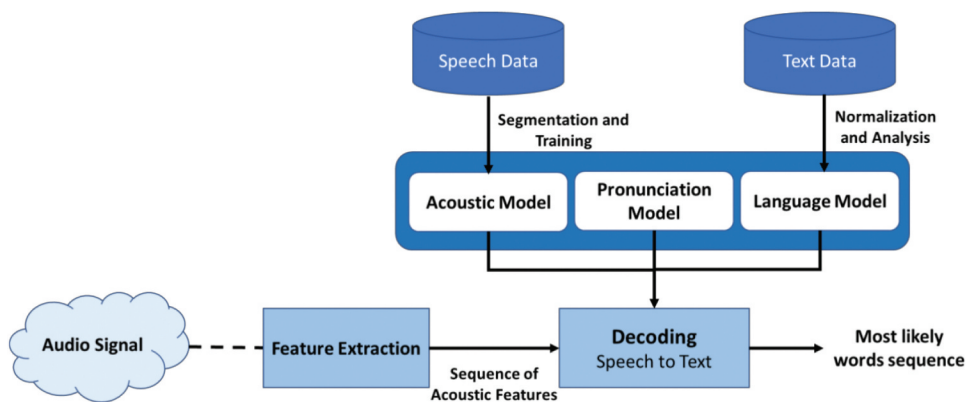


Figure 1. Architecture of an ASR.

- Linear Discriminant Analysis (LDA) and Probabilistic LDA (Ioffe 2006): This technique uses the state-dependent variables of Hidden Markov Model-based (HMM) on i-vector extraction. The i-vector is a low dimensional vector with a fixed length that contains relevant information.
- Mel-frequency cepstrum (MFCCs): It is the most commonly used technique, with a frameshift and length usually between 20 and 32 ms, using 1024 frequency bins, 26 mel channels and between 10 and 40 cepstral coefficients with cepstral mean normalization (Murphy 2012; Murshed et al., 2020; Padmanabhan & Premkumar, 2015; Renals and Grefenstette 2000). This technique has low complexity and a high ACC of recognition. Mel-Frequency Cepstrum Coefficient (MFCC) is the usual method for character extraction in most papers tackling the design of speech recognition systems for limited vocabulary (Doukas and Bardis 2017; Gerazov and Ivanovski 2013; F. Huang 2011). The public sphinx base library provides an implementation of this method that can be used directly, as was done in (X. Liu and Zhou 2014). Figure 2 provides a brief description of the MFCC method, encompassing six steps as described below.

(1) Framing and windowing: For the acoustic parameters to be stable, the speech signal must be examined over a sufficiently short period. This step aims to cut windows of 20 to 30 ms.

(2) Hamming window: The Hamming window is used to reduce the spectral distortion of the signal. This is in contrast to the Rectangular window, which is simple but can cause problems since it slices the signal boundaries abruptly; the Hamming window reduces the signal values toward zero at the boundaries. This can help avoid discontinuities. This is expressed mathematically as follows:

$$y(n) = x(n) \times w(n), 0 \leq n \leq N - 1, \quad (1)$$

where n is the number of windows, N is the number of samples in each frame, $y(n)$ the output signal, $x(n)$ the input signal and $w(n)$ the Hamming windows defined as:

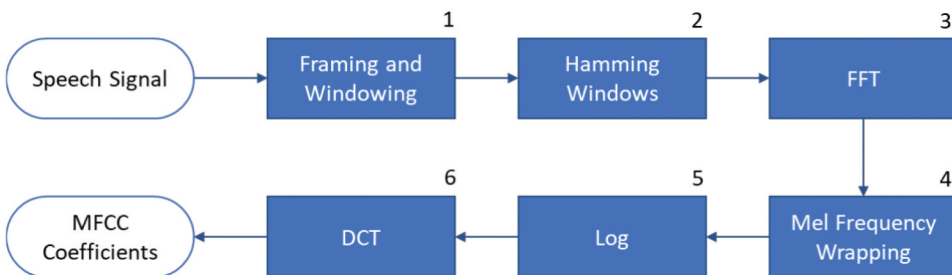


Figure 2. Block diagram of MFCC feature extraction.

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N}\right). \quad (2)$$

(3) Fast Fourier Transformation (FFT) does the conversion of time domain windows into the frequency domain by discretising and interpolating the window onto a regular grid before the Fourier transform is applied. The FFT decreases the computation requirements compared to the discrete Fourier transform. The latter is defined as:

$$S(k) = \sum_{n=0}^N y(n)e^{-2\pi K \frac{n}{N}}, 0 \leq k \leq N, \quad (3)$$

where N is the number of windows, k is the index of the coefficient and K is the number of coefficients.

(4) Mel Filter Wrapping banks allow reproduction of the selectivity of the human auditory system by providing a coefficient that gives the energy of the signal:

$$X(m) = \sum_{k=0}^{N-1} |S(k)|W(k, m), 1 \leq M, M < N, \quad (4)$$

where m is the index of the filter, M is the number of filters and W is the weight function with inputs. The k^{th} energy spectrum bin contributing to the m^{th} output band.

(5) Log allows one to obtain the logarithmic spectrum of Mel and to compress the sum $X(m)$ using:

$$X'(m) = \ln(X(m)). \quad (5)$$

(6) Discrete Cosine Transform (DCT) reduces the influence of low-energy components. MFCC coefficients are obtained by the discrete cosine transform given by:

$$c(k) = \sum_{m=0}^{N-1} a_m \cos\left[\frac{\pi}{M}\left(m + \frac{1}{2}\right)k\right] X'(m), 0 \leq k \leq K, \quad (6)$$

where $a_0 = \frac{1}{\sqrt{M}}$, $a_m = \sqrt{\frac{2}{M}}$.

After this last step, the coefficients are returned as outputs.

Acoustic Modeling

Acoustic modeling (AM) of speech typically describes how statistical representations of the feature vector sequences computed using the speech waveform are established. It aims to predict the most likely phonemes in the audio

that has been given as input (Benkerzaz, Y., & A., 2019). Challenging configurations mainly involving noise may occur. Two cases can be identified, namely the noise is isolated to a single band of frequencies, or the noise is not isolated, meaning it is on several bands. In the first case, AMs are still reliable enough to make the right decision. In the second case, models have no good information and they are prone to mistakes (Fish 2006). The Bayesian process applied to ASR is as follows: Assuming that x is a sequence of unknown acoustic vectors and $w_i (i = 1, \dots, K)$ is one of the K possible classes for an observation, the recognized class is given by:

$$w^* = \arg_i \max \frac{P(x|w_i)P(w_i)}{P(x)} \quad (7)$$

$$= \arg_i \max P(x|w_i)P(w_i). \quad (8)$$

The recognized word w^* will therefore be the one that maximizes this quantity, among all the candidate words w_i . The probability $P(x|w_i)$ of observing a signal, x , knowing the sequence needed for the AM estimation. The a prior $P(w_i)$ of the sequences is independent of the signal and needs an LM for estimation. $P(X)$ is the probability to observe the sequence of acoustic vectors X . It is the same for each phoneme sequence (because $P(X)$ does not depend on W), so it can be ignored (Renals and Grefenstette 2000).

For ASR using limited vocabulary, the most commonly used AM to estimate the probability $P(X|W)$ is the HMM (Chaudhuri, Raj, and Ezzat 2011; Gerazov and Ivanovski 2013; X. Liu and Zhou 2014; Tamgno et al. 2012). The HMM is considered a generator of acoustic vectors. It is a finite-state automaton in which the transition from state q_i to state q_j has a probability of a_{ij} at each time unit. This transition generates an acoustic vector x_t with a probability density $b_j(x_t)$. The HMM is then given by $M = (\pi_i, A, B)$, where π_i represents the initial probability distribution, A is the transition probability matrix a_{ij} , and B is the set of observation densities $B = \{b_j(x_t)\}$. Several approaches for learning HMM models have been proposed, such as the maximum likelihood estimation (Baggenstoss 2001) and forward-backward estimation (Yu and Kobayashi 2003).

Language Model

The LM in ASR is used to predict the most likely word sequence for a given text. In limited vocabulary, the training text is divided into word classes during the training phase. Based on word classes, the class-based n-gram LM is elaborated on. Thereafter, the standard bigram model, the class-based bigram model, and the interpolated model are obtained and used by the speech recognition system (Militaru and Lazar 2017).

The language statistical model (for a sequence of words $W = w_1, w_2, \dots, w_N$) consists in calculating the probability $P(W)$:

$$P(w) = \prod_{i=1}^N P(w_i | w_1, \dots, w_{i-1}) \quad (9)$$

$$= \prod_{i=1}^N P(w_i | h_i), \quad (10)$$

where $h_i = w_1, \dots, w_{i-1}$ is considered as the history of the word w_i and $P(w_i | h_i)$ is the probability of the word w_i , knowing all the previous words.

In practice, as the sequence of words h_i becomes richer, an estimation of the values of the conditional probabilities $P(w_i | h_i)$ becomes more and more difficult because no corpus of learning text can observe all possible combinations of $h_i = w_1, \dots, w_{i-1}$.

To reduce the complexity of the LM, and consequently of its learning, the n-gram approach can be used. The principle is therefore the same and only the history is limited to the previous $n - 1$ words. The probability $P(w)$ is thus approximated as:

$$P(w) = \prod_{i=1}^N P(w_i | w_{i-n+1}, \dots, w_{i-1}). \quad (11)$$

It is possible to find the probability of the occurrence of a word w_i in the learning corpus:

$$P(w_i) = \frac{C(w_i)}{C}, \quad (12)$$

where $C(w_i)$ is the number of times the word w_i has been observed in the learning corpus and C the total number of words in the corpus. In practice, depending on the size of the learning corpus, different sizes of the history can be chosen. We then speak of a unigram model if $n = 1$ (without history), a bigram if $n = 2$ or a trigram if $n = 3$.

The N-gram language model Pocket sphinx-based is used to express syntactic constraints between words. For automatic speech recognition for limited vocabulary, the online tool LMTTool of CMU is recommended to train voice data on the network server (Ashraf et al. 2010; Chaudhuri, Raj, and Ezzat 2011; X. Liu and Zhou 2014) LMTTool use a corpus in the form of an ASCII text file to make appear language model and dictionary. Always for ASR limited vocabulary, many authors use the HTK decoding parameters in language model scale factor (Al-Qatab and Ainon 2010; Alumae Alotaibi, & Huda, 2009; Qiao, Sherwani, and Rosenfeld 2010).

Pronunciation Model

The pronunciation dictionary models describe how a word is pronounced and represented. For small vocabulary, word-based models are used to define whole words as individual sound units. The pronunciation dictionary is a part of the pronunciation model. Translation of speech signal into text is achieved by classifying the speech signal into small sound units. The pronunciation model then determines how these small units can be combined to form valid words. For limited vocabularies, pronunciation models can be constructed by using hand-written word pronunciations, deriving them with phonological rules, or finding frequent pronunciations in a hand-transcribed corpus (J. Fosler-Lussier 1999). Another way to design the pronunciation model for limited vocabularies is to develop independent statistical models for each word in the dataset. The idea is to design a system in which each word has a number of parts, and then a model is trained to recognize each part of the word (E. Fosler-Lussier 2003). Once the acoustic, the language, and the pronunciation models are developed, the decoder uses them to output the text corresponding to the received audio signal.

Decoder

A decoder is seen as a graph search algorithm which combines acoustic and linguistic knowledge to automatically transcribe the input record (Benkerzaz, Y, and A 2019). The goal of decoding is to deduce the sequence of states that generated the given observations. From this sequence of states, it is easy to find the most likely sequence of phonemes that matches the observed parameters. For a limited vocabulary, the Viterbi search algorithm (Viterbi 1967) uses the probabilities of the AM and those of the LM to accomplish the decoding task. The Viterbi decoder is good for short commands, meaning for small or limited vocabulary (Hui 2019; Novak 2010). It seeks the most probable candidate among states in an HMM. This search is performed given the probability of observations obtained from the AM, for each time step for each of the states (cepstral coefficient vector corresponding to the time step).

Automatic Speech Recognition Approaches

Three techniques in artificial intelligence are used for ASR in general, namely machine learning (ML), DL, and deep reinforcement learning (DRL).

Machine Learning

Machine learning (ML) is an artificial intelligence technique that refers to systems that can learn by themselves (Myers 2019). ML implies teaching a computer to recognize patterns in contrast to the traditional approach,

which consists of programming a computer with specific rules. The teaching is done through a training process that involves feeding large amounts of audio data to the algorithm and allowing it to learn from data and detect patterns that can later be used to achieve some tasks (Murphy 2012; Murshed et al., 2020).

The different ML steps for speech recognition are provided in Figure 3 and are detailed as follows:

- (1) The first step is to select and prepare a training dataset composed of audios (from words or sentences) that have been acquired through microphones. This data will be used to feed the ML model during the learning process so that it can determine texts corresponding to audio inputs. Data must be meticulously prepared, organized and cleaned with the aim to mitigate bias during the training process.
- (2) The second step is to perform a pre-processing on the input data. This pre-processing includes the reduction of the noise in the audio and the enhancement of data.
- (3) The third step consists of choosing a parametric class where the model will be searched. This is a fine-tuned process that is run on the training dataset. For speech recognition using limited vocabulary, some algorithms such as the Maximum Likelihood Linear Regression algorithm (X. Liu and Zhou 2014) can be used to train the AM, and the Viterbi algorithm (F. Huang 2011; X. Liu and Zhou 2014) for decoding. The type of algorithm to use depends on the type of problem to be solved.
- (4) The fourth step is the training of the algorithm. This is an iterative process. After running the algorithm, the results are compared with the expected ones. The weights and biases are eventually tuned via the back propagation optimization, to increase the accuracy of the algorithm. This process is repeated until a certain criterion is met and the resulting trained model is saved for further analysis with the test data.
- (5) The fifth and final step is to use and improve the model. The model is then used on new data.



Figure 3. ML technique.

Different ML methods have been used for acoustic modeling in speech recognition systems (Padmanabhan and Premkumar 2015). The evaluation, decoding and training of HMMs are done by ML forward-backward (Yu and Kobayashi 2003), Viterbi (Viterbi 1967) and Baum-Welch algorithms (Baggenstoss 2001), respectively. In their work, Padmanabhan et al. (Padmanabhan and Premkumar 2015) review these methods.

Deep Learning

Deep Learning (DL) is a set of algorithms in ML. It uses model architectures made up of multiple non-linear transformations (neural networks) to model high-level abstractions in data (Jinyu et al. 2016). Deep Neural Networks (DNNs) work well for ASR when compared with Gaussian Mixture Model-based HMMs (GMM-HMM) systems, and they even outperform the latter in certain tasks (Hinton et al. 2012). DL employs the Convolutional Neural Network (CNN) approach which owns the ability to automatically learn the invariant features to distinguish and classify the audio (Abhishek 2017).

By learning multiple levels of data representations, DL can derive higher-level features from lower-level ones to form a hierarchy. For instance, in a speech classification task, the DL model can take phoneme values in the input layer and assign labels to the word in the sentence in the output layer. Between these two layers, there are a set of hidden layers that build successive higher-order features that are less sensitive to conditions, such as noise in the user's environment (Hernández-Blanco, Herrera-Flores, Tomás, & NavarroColorado, 2019).

DL can be implemented using various tools. However, Tensor Flow seems to be one of the best application methods currently available (Dhankar 2017). [Figure 4](#) gives the steps of DL in ASR. Data augmentation helps to improve the performance of the model by generalizing better and thereby reducing overfitting (Salamon and Bello 2017). Data augmentation creates a rich, diverse set of data from a small amount of data. Data augmentation can be applied as a pre-processing step before training the model or later, directly in real-time. Different augmentation policies can be applied to audio data such as Time warping, Frequency masking, and Time masking. Recently, a new augmentation method called SpecAugment has been proposed by Park et al. in (Park 2019) for the ASR system. They combined the warping of the features and the masking of blocks of frequency channels, as well as the blocks of time steps. To ease the augmentation process, a recent free MATLAB Toolbox called Audiogmenter has been proposed (Maguolo et al. 2019).

The feature extraction process aims to remove the non-dominant features and therefore reducing the training time while mitigating the complexity of the developed models.

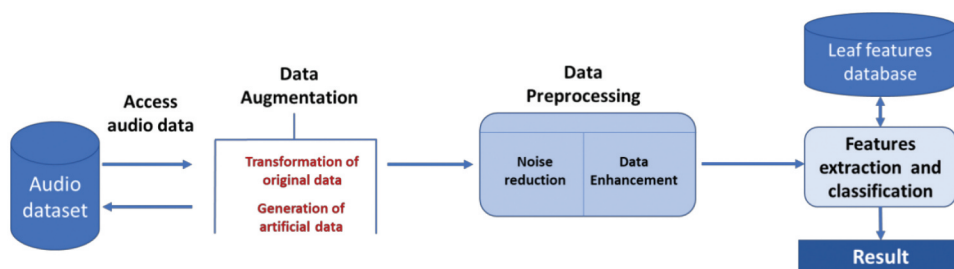


Figure 4. Steps of DL.

Deep Reinforcement Learning

A speech recognition system can be vulnerable to a noisy environment. To address this issue, deep reinforcement learning (DRL) can achieve complex goals in an iterative manner, which makes it suitable for such applications. Reinforcement learning is a popular paradigm of ML, which involves agents learning their behavior by trial and error. DRL is a combination of standard reinforcement learning with DL to overcome the limitations of reinforcement learning in complex environments with large state spaces or high computation requirements. DRL enables software-defined agents to learn the best actions possible in virtual environments to attain their goals (Mnih, Kavukcuoglu, and Silver 2020). This technique has recently been applied to limited vocabulary such as the “Speech Command” dataset in (Rajapakshe et al. 2020) or larger vocabulary such as (Kala and Shinozaki 2018). Regardless of the artificial intelligence technique that is used, an important prerequisite remains, namely the dataset.

Datasets Management

Construction

Speech recognition research has traditionally required the resources of large organizations, such as universities or corporations (Warden 2018). Microphones recording can save data coming from multiple sources; this permits the collection of enough data for recognition. Although this approach remains the most frequent use case, it is subject to some challenges including speaker localization, speech enhancement, and ASR in distant-microphone scenarios (Cohen, Benesty, and Gannot 2010; Vincent, Virtanen, & Gannot, 2018). Datasets with limited vocabulary may be a subset of larger speech datasets (Glasser 2019). Datasets are used to train and test ASR engines. It is important to have a multi-speakers database (Gerazov and Ivanovski 2013). To avoid inconsistencies between datasets, the collection is done during a short period (the same day if possible) (Hofe, 2013). For the best results, the corpus of acoustic data used for learning must be performed in a good

quality recording studio, but it does not have to occur in a professional studio. For instance, authors in (Tamgno et al. 2012) used a Handy Recorder with four channels (H4n) that is affordable. The creation of an audio dataset goes through four steps as shown in Figure 5. These steps are: requirement definition, corpus creation, voice recording, and labeling of the voice database.

a.) Requirement definition: The first step consists of selecting a suitable environment, which can be a room with closed doors to reduce noise. The use of studio recorded audio is unrealistic, as these audios are free of background noise and recorded with high-quality microphones and are in a formal setting. Good ASR models should work equally well in moderated noisy environments with natural voices. For good recognition, audio recordings should be made with several people (female and male) having different tones. The data should have a fixed and short duration to facilitate the training of the learning model and the evaluation process (Warden 2018).

b.) Corpus creation: After requirement definition, words or sentences are chosen. The choice is made according to the needs, and it is limited to the context in which speech recognition will be performed. Only necessary data should be selected.

c.) Voice recording: Sounds are recorded using a microphone that matches the desired conditions. It should minimize differences between training conditions and test conditions. Speech recordings are generally performed in an anechoic room and are usually digitized at 20 kHz using 16 bits (Alumae and Vohandu 2004; Glasser 2019) or at 8 kHz (Tamgno et al. 2012) or 16 kHz (Hofe et al. 2013; Warden 2018). The waveform audio file format container with file extension .wav is generally used (Glasser 2019; Warden 2018). The WAV formats encoded to Pulse-Code Modulation (PCM) allow one to obtain an uncompressed and high-fidelity digital sound. Since these formats are easy to process in the pre-processing phase of speech recognition and for further processing, it is necessary to convert the audio files obtained after the recording (for instance OGG, WMA, MID, etc.) into WAV format.

d.) Labeling of the voice database: Most ML models are done in a supervised approach. For supervised learning to work, a set of labeled data from which the model can learn to make correct decisions is required. The labeling step aims to identify raw data (text and audio files) and to add informative labels in order

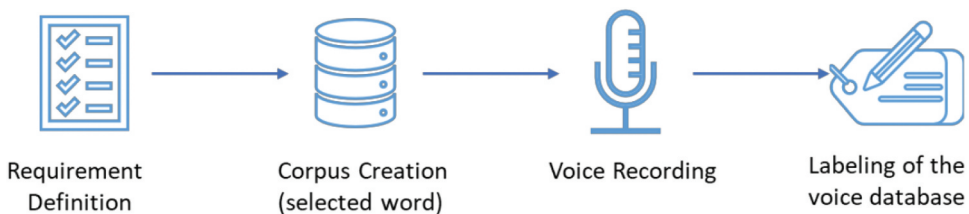


Figure 5. Construction of dataset for limited vocabulary.

to specify the context so that an ML model can learn from it. Each recorded file is marked by sub-words or phonemes. Labeling indicates which phonemes or words were spoken in the audio recording. To label the data, humans make judgments about some aspects of the unlabeled audio file. For example, one might ask to label all audios containing a given word or phoneme. The ML model uses human-supplied labels to learn. The resulting trained model is used to make predictions on new data. Some software such as Audio Labeler allows one to define reference labels for audio datasets. Audio Labeler also allows one to visualize these labels interactively.

The minimal structure of a dataset for limited vocabulary takes into account elements such as the path to the audio file, the text corresponding to the audio file, the gender of the speaker, the age and the language used if there are many languages in the dataset.

If there are several speakers, then the speaker's index will be one key. Some researchers also take into account the emotion of the speaker, mentioning if he is neutral, happy, angry, surprised or sad (Zhou et al. 2021).

Languages and Datasets

In the literature, several works have developed datasets such as VoiceHome2 (Bertin, 2019) that developed a French corpus for distant microphone speech processing in domestic environments. Also regarding French, the work in (Mezzoudj et al. 2018) proposes a multi-source data selection for the training of LMs dedicated to the transcription of broadcast news and TV shows. An important work focusing on the English language is the reduced voice command database (Pleshkova, Bekyarski, and Zahariev 2019). It has been created from a worldwide cloud speech database and in combination with training, testing and real-time recognition algorithms based on artificial intelligence and DL neural networks. Another important English dataset for digits from 0 to 9, and 10 short phrases is the AV Digits Database (Petridis et al. 2018). In a survey, 53 participants consisting of 41 males and 12 females were asked to read digits in English in random order five times. In another study, 33 agents were asked to record sentences, each sentence was repeated five times in 3 different modes: neutral, whisper and silent speech. A larger dataset is the Isolet dataset proposed in (Asuncion and Newman 2007). This dataset contains 150 voices divided into five groups of 30 people. In this dataset, each speaker pronounces each letter of the English alphabet twice, which provides a set of 52 training examples for each speaker.

Apart from French and English, other languages, such as Sorani Kurdish have been tackled. BD-4SK-ASR (Basic Dataset for Sorani Kurdish Automatic Speech Recognition) is an experimental dataset which is used in the first attempt in developing an ASR system for Sorani Kurdish (Qader and Hassani 2019).

A very large project run by Mozilla is the Common Voice Project initiated with the aim of producing an open-source database for ASR (Ardila et al. 2020). The Mozilla Common voice dataset created in 2017 is intended for developers of language processing tools. In November 2020, more than 60 languages were represented on the platform. These languages include French, English, Chinese, Danish and Norwegian. It is very important that speech recognition systems be tested for efficiency, regardless of the language.

Performance Metrics

The quality of the output transcript of an ASR system is traditionally measured by the word error rate (WER) metric:

$$WER = \frac{S + I + D}{N}, \quad (13)$$

where S is the number of incorrect words substituted, I is the number of extra words inserted, D is the number of words deleted and N is the number of words in the correct transcript. The WER metric is estimated in terms of percentage. However, it is important to note that it is possible to have a WER value exceeding 100% and the WER threshold for acceptable performance depends on the applications. However, Johnson et al. (Johnson et al. 1999) have shown that it is still possible to get a good retrieval performance with a WER value up to 66%. However, the precision begins to fall off quickly when WER gets above 30% (Johnson et al. 1999). Having an estimation of WER value and knowing the threshold for usable transcripts, the allocation of processing resources can be oriented only to those files predicted to yield usable results. If a spoken word is not included in the vocabulary, the corresponding transcript will be considered as an error and the WER value will increase.

For detection of keywords, authors in (Anh and Thi 2020) proposed a new method to calculate the mean of accuracies of each keyword (acc_i). This is computed depending on the number of keywords correctly predicted (N_{cp}), the number of keywords not yet predicted (N_{ny}) and the number of keywords incorrectly predicted (N_{ip}):

$$acc_i = \frac{N_{cp} - N_{ip}}{N_{cp} + N_{ny}}. \quad (14)$$

The ACC is computed from the above as:

$$ACC = \frac{1}{N} \sum_{i=0}^{N-1} acc_i. \quad (15)$$

The WER metric works well when the morphology of the language is simple (Besacier et al. 2014). Otherwise, more adequate metrics should be applied, such as the Letter or Character Error Rate (LER or CER) (Kurimo et al. 2006), the Syllable Error Rate (SyLER) (C. Huang et al. 2000) or Speaker Attributed Word Error Rate (SA-WER) (Galibert 2013).

Use of OPD (Output Probability Distributions) and secondary classification is a solution to improve accuracy of ASR isolated word in limited vocabulary (Thambiratnam and Sridharan 2000). To do this, it models the relationships between words. OPD represents the distribution of logarithm probability of HMM's set. For each word of vocabulary, a HMM is trained, an utterance is transferred to HMM and logarithmic probabilities are concatenated to give OPD.

The LMs can be evaluated separately from the AM and the most commonly used measure is perplexity (Jelinek et al. 1977). This measure is calculated on a text not seen during training (test-set perplexity). Another measure is based on Shannon's game (Shannon 1951). Existing frameworks are used to develop ASR systems. Comparing these and previous works can enable newcomers in the field to develop ASR systems.

Speech Recognition Frameworks or Toolkits

Several toolkits for ASR have been developed to train datasets. Some toolkits are open-source code and others are not and their presentation is the aim of this section.

Closed-source Code Systems

Several closed-source systems are available for ASR, namely the Dragon Mobile software development kit (SDK), Google Speech Recognition API, SiriKit, Yandex SpeechKit and Microsoft Speech API (Matarneh et al. 2017).

The Dragon Mobile SDK, developed by Nuance since 2011, provides speech services to enhance applications with speech recognition and text-to-speech functionality. It consists of a set of sample projects, documentation, and a framework to ease the integration of speech services into any application. It is a trialware that requires a paid subscription after 90 days. Popular mobile platforms are supported (Android, iOS, and Windows phone). It has been used in some works like (Fujiwara 2016) where a custom phonetic alphabet has been optimized to facilitate text entry on small displays.

The Yandex SpeechKit can be used to integrate speech recognition, text-to-speech, music identification, and Yandex voice activation into Android mobile applications. The Yandex SpeechKit supports the following languages for

speech recognition and text-to-speech: Russian, English, Ukrainian and Turkish. Yandex SpeechKit has been recently used in (Prozorov and Tatarinova 2019).

SiriKit is a toolkit that allows one to integrate Siri into a third-party iOS or macOS application, to benefit from the ASR capabilities. Siri is a speech recognition personal assistant unveiled by Apple in 2011 and only works on iOS and macOS (Rawat, Gupta, and Kumar 2014). Even though potential dangers and ethical issues have been reported (Zeng 2015), SiriKit is used in numerous works including (Herbert and Kang 2019).

The Speech Application Programming Interface (SAPI) developed by Microsoft enables developers to integrate speech recognition and speech synthesis within Windows applications (Gaida et al. 2014). Several versions of the API have been released either as part of a Speech SDK or as part of the Windows Operating System. SAPI is used in Microsoft Office, Microsoft Agent and Microsoft Speech Server. Several research works have also made use of this API (Shi and Maier 1996).

Google Speech Recognition API is a C# toolkit based on a model trained with English and 120 other languages. Google has improved its speech recognition by using a DNN in its applications, reaching an 8% error rate in 2015 compared to 23% in 2013 (Kěpuska 2017).

Table 2 provides a summary of selected closed-source code speech recognition toolkits.

As conventional closed-source software, these toolkits are not flexible and are limited to the languages they have been designed for. Such limitations fostered the research community to develop open-source frameworks and toolkits.

Open-source Code Systems

A plethora of open-source frameworks, engines, or toolkits for ASR systems are proposed in the literature. The following is a non-exhaustive list of main works or projects. Most of them can easily handle small and large vocabulary. The HTK is implemented in the late 1980s, and maintained by the Speech Vision and Robotics Group of the Cambridge University Engineering Department (CUED) (Young 2002); HTK is available to the research community since early 2000. It provides recipes to build baseline systems with HMM. HTK is considered a very simple and effective tool for research (Qiao, Sherwani, and Rosenfeld 2010; Supriya and Handore 2017). It can build a noise-robust ASR system in a moderated noisy level environment, especially for small vocabulary systems. It is a practical solution to develop fast and accurate Small Vocabulary Automatic Speech Recognition (SVASR) (Hatala 2019). One of the most popular toolkits is the CMU Sphinx, designed for both mobile and server applications.

Table 2. A non-exhaustive list of closed-source code speech recognition frameworks/toolkits.

Toolkit/1 st Release	Programming Language	Targeting language	Applications	Applied technology
Microsoft Speech API, 1995	C#	English, Chinese, Japanese	Speech recognition, speech synthesis within windows applications	Context-dependent DNN, HMM
SiriKit, 2011	Objective-C	English, German, French	Speech recognition	DNN and ML
Dragon Mobile SDK, 2011	C++	English, Spanish, French, German, Italian, Japanese	Speech recognition and text to speech	Markov processes
Yandex Speechkit, 2014	C, Python	Russian, English, Ukrainian, Turkish	Speech recognition, text to speech, music identification and voice activation	Neural network
Google Speech Recognition API, 2016	C#	English, more than 120 languages	Speech to text, speech recognition	DL neural network

CMU Sphinx is in fact a set of libraries and tools that can be used to develop speech-enabled applications. It is developed at Carnegie Mellon University in the late 1980s (Lee 1988). Several versions have been released including Sphinx 1 to 4, and PocketSphinx for hand-held devices (Huggins-Daines et al. 2006). CMU Sphinx is currently attracting the attention of the research community. It offers the possibility to build new LMs using its language Modeling Tool.

A former toolkit that may no longer be available is the Rapid Language Adaptation Toolkit (RLAT) introduced in (Schultz 2009) and used in (Vu et al. 2010). The website for the project is, unfortunately, no longer available. Kaldi is an extendable and modular toolkit for ASR (Povey 2011). The large community behind the project provides numerous third-party modules that can be used for several tasks. It supports DNN and offers excellent documentation on its website. Several works have been based on this toolkit including (Guglani and Mishra 2018).

Microsoft also proposes an open-source Cognitive Toolkit (CNTK). It is used to create DNNs that power many Microsoft services and products. It enables researchers and data scientists to easily code such neural networks at the right abstraction level, and to efficiently train and test them on production-scale data (Banerjee, Hamidouche, and Panda 2016). Although it is still used, it is a deprecated framework.

Julius is software normally designed for LVCSR. It is based on word N – *gram* and context-dependent HMM. It is used in several works including (Sharma et al. 2019).

Simon toolkit is a general public license speech recognition framework developed in C++. It is designed to be as flexible as possible and it works with any language or dialect. Simon makes use of KDE libraries, CMU SPHINX or Julius together with HTK and it runs on Windows and Linux.

Praat is a framework that enables speech analysis, synthesis, and manipulation. In addition, it allows speech labeling and segmentation. It has been used in (Pleva, Juhár, and Thiessen 2015).

Mozilla Common Voice is a free speech recognition software for developers that can be integrated into projects. It works with DNN technology and targets several languages (Ardila et al. 2020). Besides Common Voice, Mozilla has also developed DeepSpeech, an open-source Speech-To-Text engine. It makes use of a model trained by the ML techniques proposed in (Hannun 2014).

Another larger project is the OpenSMILE (open-source Speech and Music Interpretation by Large-space Extraction) project that is completely free to use for research purposes. It received a lot of attention from the research community and claims more than 150,000 downloads. A recent model called Jasper (Just Another Speech Recognizer) has been introduced in 2019 (Li 2019). It can be used with the OpenSeq2Seq TensorFlow-based toolkit. OpenSeq2Seq enables, among others, speech recognition, speech commands and speech synthesis. It is used in recent works such as (S. Liu et al. 2021).

Other recent engines for ASR have been released such as Fairseq and Wav2Letter++ (both developed by Facebook), Athena, ESPnet, and Vosk which is an offline ASR toolkit. Table 3 provides a summary of some open-source speech recognition toolkits.

Several works have been performed for limited vocabularies in ASR using the above toolkits. We make a summary of these works in the following section.

Summary of Related Work on ASR Using Limited Vocabulary

Selected Works on ASR Using Limited Vocabulary

ASR using limited vocabulary has attracted the attention of researchers. The literature proposed systems for European languages such as English (Chaudhuri, Raj, and Ezzat 2011; Y. Liu et al. 2015; Lucey, ran, and Chandran 2001), Estonian (Alumae and Vohandu 2004), Norwegian (Haflan 2019), Romanian (Burileanu et al. 2003; Militaru and Lazar 2017), Macedonian (Gerazov and Ivanovski 2013), as well as Asian languages including Chinese (F. Huang 2011; X. Liu and Zhou 2014), Hindi (Bali et al. 2013), Indonesian (Hatala 2019), Arabic (Al-Qatab and Aion 2010; Azmi and Tolba 2008), Bangla (Muhammad, Alotaibi, and Huda 2009), Malayalam (Krishnan, Jayakumar, and Babu 2008; Sukumar, Shah, and Anto 2010), Tamil (Rathinavelu and Anupriya 2007), Marathi (Supriya and Handore 2017),

Table 3. A non-exhaustive list of open-source speech recognition frameworks/toolkits.

Toolkit/1 st Release	Programming language	License	Trained models	Applied technology
Praat, 1991	C/C++, Objective C	GNU General Public License	-	Neural networks
HTK, 2000	C/C++	HTK Specific License	English	HMM, DNN
CMU Sphinx, 2001	Java	BSD License	English plus 10 other languages	HMM
Simon, 2008	C++	GNU General Public License	-	HMM
Kaldi, 2009	C++	Apache	Subset of English	DNN
OpenSmile, 2010	C++	Free for noncommercial use	-	Machine/DL
Wav2Letter, 2016	Torch (Lua)	BSD License	English	Neural Networks
Microsoft CNTK, 2016	C++	BSD License	-	DNN
Mozilla Common Voice, 2017	C, Python, Java	Public domain CC-0	English, French, German, Chinese, ...	DNN
OpenSeq2Seq, 2018	Python	Apache	-	DNN
Fairseq, 2019	Python	BSD License	English, Chinese, German, and French	Neural Networks

Punjabi (Mittal and Singh 2020) and Urdu (Ashraf et al. 2010; Benisty et al. 2018). Table 4 provides a list of selected works done on ASR using limited vocabulary base.

Remarks

Even though European and Asian languages are well represented in the selected works, we noticed a particular focus on well-resourced languages, namely English, Chinese, Romanian. Only a few works deal with African languages namely (Tamgno et al. 2012) and Yoruba (Qiao, Sherwani, and Rosenfeld 2010). The weak representation of African languages can be justified by the fact that most of them are under-resourced, in addition to the lack of local skills and awareness about the potential of ASR systems to prevent the extinction of under-resourced languages. Although most of the works did consider noisy environments, the severity was moderated and usually limited to natural environmental noises. Due to the limited size of the vocabulary, such noise levels have been easily mitigated. This justified the competitive ACC in most works.

We observe that the HTK Toolkit is the most used for speech recognition with regards to limited vocabulary. Its success is explained by the fact that it eases the manipulation of HMM parameters for the training and testing stage of system development.

In general, the Bayesian equation allows one to find the probability that a word will be recognized in the speech recognition process for limited vocabulary, and in turn the sequence of words corresponding to a speech

Table 4. Summary of works on ASR using limited vocabulary.

Papers, Year	Language	Toolkit/Model	Noisy	Acc (%)	# Speakers	Dataset
(Lucey, ran, and Chandran 2001)	English	HTK	Yes	-	10	780 utterances
(Burlileanu et al. 2003)	Romanian	LGB and LVQ3	Yes	95.33	30	10 Digits
(Alumae and Vohandu 2004)	Estonian	HTK	No	97	60	400 utterances of numbers
(Rathinavelu and Anupriya 2007)	Tamil	Feedforward neural networks	Yes	81	10	20 phonemes
(Krishnan, Jayakumar, and Babu 2008)	Malayalam	From scratch using ANN	Yes	89	-	5 words
(Azmi and Tolba 2008)	Arabic	HTK	Yes	81.79	59	944 sentences
(Muhammad, Alotaibi, and Huda 2009)	Bangla	HTK	Yes	90–95	100	10 digits
(Qiao, Sherwani, and Rosenfeld 2010)	Yoruba/ Hebrew	HTK	No	100	3	50 words/ short phrases
(Ashraf et al. 2010)	Urdu	CMU Sphinx	Yes	60	10	52 words
(Al-Qatab and Ainon 2010)	Arabic	HTK	No	97.99	13	33 words
(Sukumar, Shah, and Anto 2010)	Malayalam	From scratch: DWT and ANN	Yes	80	1	250 words
(Chaudhuri, Raj, and Ezzat 2011)	English	CMU Sphinx	No	71.73	-	11 words
(F. Huang 2011)	Chinese	From scratch and HMM-based	Yes	89.6	8	640 speech samples
(Tamgno et al. 2012)	Wolof	HTK	Yes	81.9	25	Digits
(Bali et al. 2013)	Hindi	Microsoft Speech Server	Yes	90	24	79 words
(Gerazov and Ivanovski 2013)	Macedonian	HTK	Yes	94.52	30	188 words
(X. Liu and Zhou 2014)	Chinese	PocketSphinx (CMU)	Yes	90	1	10 voice commands
(Y. Liu et al. 2015)	English	Kaldi	Yes	46.4	20	256 words
(Supriya and Handore 2017)	Marathi	HTK	No	80–90	5	620 words
(Militaru and Lazar 2017)	Romanian	SRILM toolkit, ProtoLOGOS	No	59.57	30	762 phrases
(Benisty et al. 2018)	English	Kaldi	Yes	98	-	62 keywords
(Haflan 2019)	Norwegian	CMU Sphinx	Yes	58.31	24	10 sentences
(Hatala 2019)	Indonesian	HTK	Yes	100	1	Digits
(Barkani, Satori, Hamidi, Zealouk, & Laaidi, 2020)	Amazigh	CMU Sphinx	No	92.22	30	13 commands
(Mittal and Singh 2020)	Punjabi	CMU Sphinx	No	69.6	50	91 words
(Jongman, Khoe, and Hintz 2021)	Dutch	PPVT	Yes	75.97	132	384 words
(Yi et al. 2021)	Chinese	Keras/Tensorflow	Yes	97.5	-	877 words
(Tejedor-García, Cardeñoso- Payo, & Escudero-Mancebo, 2021)	English	Google's Speech-to-Text API	Yes	95.2	432	300 shorts phrases
(Adnene, Sabri, and Mohammed 2021)	Spanish	Kaldi	Yes	55.78	33	164 words
	English	Labview and NI my RIO 1900	Yes	96.375	3	10 words

sequence. HMM and DNN are decent AMs that achieve good speech recognition results. HMM is used to account for variability in speech and DNN, with many hidden layers, have been shown to outperform GMMs on a variety of speech recognition benchmarks. The construction of datasets for ASR with limited vocabulary is done with speech recorded and digitized at varying frequencies between 8 kHz and 20 kHz in most cases. Finally, the ACC of the system is calculated by using the number of words correctly predicted, the

number of words not yet predicted, and the number of words incorrectly predicted. ACC of ASR for limited vocabulary for previous works is between 46 and 100%.

Future Directions

The World has more than 7000 languages according to the Ethnologue website,¹ with the majority being under-resourced and even endangered. ASR systems offer an unprecedented opportunity to sustain such under-resourced languages and to fight the extinction of endangered ones. We should differentiate two types of under-resourced languages: those with an acknowledged written form and those without. In the first case, new datasets should be created and require new approaches for data recording and labeling, especially when there are not enough native speakers during the creation process. This is a challenge, especially with tonal languages. In fact, most languages in the developing world and especially in Sub-Saharan Africa are tonal (Downing & Ri- alland, 2016). In a recent survey on ASR for tonal languages (Kaur, Singh, and Kadyan 2021), only two African languages were reported. In addition, some languages (especially dialects) share commonalities. Therefore, ASR systems with limited vocabulary targeting multiple similar languages can be designed.

In the second case, when the language that does not have an acknowledged written form, a new approach should be designed. Normally, an ASR system aims to transcribe a speech into a text. In ASR using a limited vocabulary, the transcript text is usually a command or a short answer that can be used by an application or system to perform an action. In this scenario, the system can be a combination of a Speech-to-Speech Translation (S2ST) and a Speech Recognition. First, the speech in a non-written language is directly translated into a speech in a well-resourced language such as English, then the transcript in a well-resourced language is retrieved (generated) and sent to the application. The Direct S2ST model has been developed in (Jia 2019), translating Spanish into English without passing through text. Their dataset is a subset of the Fisher dataset and is composed of parallel utterance pairs. The construction of datasets for ASR systems using limited vocabulary of nonwritten languages can also be based on the same principle.

Regardless of whether the language is written or not, more noise resistant models should be developed, because the available data for under-resourced languages could be of low quality.

The limited computing resources and poor internet connectivity in some regions can prevent the use of ASR systems. There is currently a shift of ASR models from the cloud to the edge. It is performed by reducing the size of models and making models fast enough so that they can be executed on typical

mobile devices. The latest optimization techniques to achieve this, such as pruning, efficient Recurrent Neural Network variants and quantization, are presented in (Shangguan et al. 2019). Although they provide remarkable results, such as reducing the size by 8.5x and increasing the speed by 4.5x, there is still a need to develop light and offline models that can be deployed on low-resource devices, such as off-the-shelf smartphones or raspberry Pi/Arduino modules.

Conclusion

This paper presented a review of ASR systems with a focus on limited vocabulary. After introducing the ASR principle in general and usual techniques used to perform recognition, this paper also discussed the management of datasets, the performance metrics to evaluate ASR systems, and toolkits to develop such systems. From the analysis of selected papers on ASR using limited vocabulary, HMM and DNN-based AMs achieve good speech recognition results. DNN even outperform GMMs on a variety of speech recognition benchmarks. Datasets with limited vocabulary are constructed with speech at frequencies between 8 kHz and 20 kHz. The evaluation of systems is mainly based on the ACC rather than the WER metric. Despite the satisfactory results, there is still much to do. In fact, developed systems deal mostly with well-resourced languages and most models are still running on servers. We hope that the ideas for future directions discussed regarding under-resourced/unwritten languages (investigating direct speech to speech translation) and designing “on the edge ASR systems for limited vocabulary” will draw the attention of the research community to develop systems especially for developing world, where the limitation in terms of computing resources coupled with the lack of connectivity constitute quite significant barriers.

Note

1. <https://www.ethnologue.com>


Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Jean Louis K. E Fendji  <http://orcid.org/0000-0001-9803-6981>

Blaise O. Yenke  <http://orcid.org/0000-0001-9670-1001>

Marcellin Atemkeng  <http://orcid.org/0000-0002-9020-3885>

References

- Abdul-Kader, S., and D. J. Woods. 2015. Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications* 6 (7).
- Abhishek, D. 2017. Study of deep learning and cmu sphinx in automatic speech recognition. In *International conference on advances in computing, communications and informatics (icacci)*. Udupi: 2296–301.
- Adnene, N., B. Sabri, and B. Mohammed (2021). Design and implementation of an automatic speech recognition based voice control system. In *Conference on electrical engineering 2021*. (EasyChair Preprint, Conference on Electrical Engineering 2021.)
- Agnes, M., and D. Guralnik. 1999. *Webster's new world college dictionary*.
- Albabbain, A., D. AlMulhim, F. Yunus, and M. Househ. 2014. The role of mobile health in the developing world: A review of current knowledge and future trends. *Cyber J. Multidiscip. J. Sci. Technol. JSHI J. Sel. Areas Health, Informatics* (42):10–15.
- Al-Qatab, B., and R. Ainon. 2010. Arabic speech recognition using hidden markov model toolkit(htk). In *2010 international symposium on information technology, Kuala Lumpur*. Malaysia: 557–62.
- Alumae, T., and L. Vohandu. 2004. Limited-vocabulary Estonian continuous speech recognition system using hidden Markov models. *Informatica, Institute of Mathematics and Informatics* 15 (3):303–14.
- Anh, N., and H. Thi. 2020. Kwa: A new method of calculation and representation accuracy for speech keyword spotting in string results. *International Journal of Advanced Computer Science and Applications* 11 (2). doi:10.14569/IJACSA.2020.0110283.
- Ardila, R., M. Branson, K. Davis, M. Henretty, M. Kohler, and J. Meyer. 2020. *Common voice: A massively-multilingual speech corpus*. *ArXiv Prepr. ArXiv191206670*.
- Ashraf, J., N. Iqbal, N. Khattak, and A. Zaidi. 2010. *Speaker independent urdu speech recognition using hmm*. In *Natural language processing and information systems*, vol. 6177, 140–48. Berlin, Heidelberg: Springer.
- Asuncion, A., and D. Newman. 2007. *Uci machine learning repository*. Irvine, CA, USA.
- Azmi, M., and H. Tolba. 2008. Syllable-based automatic Arabic speech recognition in different conditions of noise. In *2008 9th international conference on signal processing*. Beijing, China: 601–04.
- Bagginstoss, P. 2001. A modified Baum-welch algorithm for hidden Markov models with multiple observation spaces. *IEEE Transactions on Speech and Audio Processing* 9 (4):411–16. doi:10.1109/89.917686.
- Bali, K., S. Sitaram, S. Cuendet, and I. Medhi. 2013. *A hindi speech recognizer for an agricultural video search application*. In *Proceedings of the 3rd acm symposium on computing for development - acm dev*, vol. 13, 1. Bangalore, India.
- Banerjee, D., K. Hamidouche, and D. Panda (2016). Re-designing cntk deep learning framework on modern GPU enabled clusters. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (p. 144–51).
- Barkani, F., H. Satori, M. Hamidi, O. Zealouk, and N. Laaidi (2020). Amazigh speech recognition embedded system. In *2020 1st international conference on innovative research in applied science, engineering and technology, Meknes and Morocco, APR* (p. 1–5).
- Benisty, H., I. Katz, K. Crammer, and D. Malah. 2018. Discriminative keyword spotting for limited-data applications. *Speech Communication* 99:1–11. doi:10.1016/j.specom.2018.02.003.
- Benkerzaz, S., E. Y., and D. A. 2019. A study on automatic speech recognition. *Journal of Information Technology Review* 10 (3):10.

- Bertin, N., E. Camberlein, R. Lebarbenchon, E. Vincent, S. Sivasankaran, I. Illina, and F. Bimbot. 2019. Voicehome-2, an extended corpus for multichannel speech processing in real homes. *Speech Communication* 106:68–78. doi:10.1016/j.specom.2018.11.002.
- Besacier, L., E. Barnard, A. Karpov, and T. Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication* 56:85–100. doi:10.1016/j.specom.2013.07.008.
- Burileanu, D., M. Sima, C. Negrescu, and V. Croitoru. 2003. Robust recognition of small vocabulary telephone quality speech. *Proceedings of the second conference on speech technology and human-computer dialogue (sped 2003) Bucharest, Romania*.
- Chaudhuri, S., B. Raj, and T. Ezzat. 2011. A paradigm for limited vocabulary speech recognition based on redundant spectro-temporal feature sets. *INTERSPEECH* 4.
- Clements, G. 1985. The geometry of phonological features. *Phonology Yearbook* 2 (1):225–52. doi:10.1017/S0952675700000440.
- Cohen, I., J. Benesty, and S. Gannot. 2010. *Speech processing in modern communication: Challenges and perspectives*. Berlin: Springer.
- Desai, N., K. Dhameliya, and V. Desai. 2013. Feature extraction and classification techniques for speech recognition: A review. *International Journal of Emerging Technology and Advanced Engin E Ering* 3 (5).
- Dhankar, A. (2017). Study of deep learning and cmu sphinx in automatic speech recognition. In *2017 international conference on advances in computing, communications and informatics (icacci)* (p. 2296–301). Udupi.
- Doukas, N., and N. Bardis. 2017. Current trends in small vocabulary speech recognition for equipment control. In *Mathematical methods and computational techniques in science and engineering*. Cambridge, UK: 020029.
- Doukas, N., N. Bardis, and O. Markovskiy (2017). Task and context aware isolated word recognition. In *2017 international conference on control, artificial intelligence, robotics and optimization (iccairo)* (p. 330–34).
- Downing, L., and A. Rialland. 2016. *Intonation in African tone languages*, vol. 24. Walter de Gruyter GmbH and Co KG.
- Ebongue, J. 2015. Rethinking network connectivity in rural communities in Cameroon. *Ist-Africa*. Lilongwe, Malaw. Retrieved from <https://arxiv.org/ftp/arxiv/paper/1505/1505.04449.pdf> (Online]. Available:)
- Ebongue Louis, F. (2015). *Wireless mesh network: a rural community case*. Universitat Bremen. Retrieved from <https://www.researchgate.net/publication/282297986/Wireless/Mesh/Network/a/rural/community/case> (Accessed: Oct. 20, 2020. [On- line]. Available:)
- Fish, R. (2006). Using audio quality to predict word error rate in an automatic speech recognition system. In (p. 4). Retrieved from <https://www.mitre.org/sites/default/files/pdf/06/1154.pdf>
- Fosler-Lussier, E. 2003. A tutorial on pronunciation modeling for large vocabulary speech recognition. In *Text- and speech-triggered information access*, ed. S. Renals and G. Grefenstette., vol. 2705, 38–77. Berlin, Heidelberg: Springer.
- Fosler-Lussier, J. 1999. Dynamic pronunciation models for automatic speech recognition. *International Computer Science Institute* 204. Retrieved from <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.42.7647/&rep=rep1/&type=pdf>
- Fujiwara, K. (2016). Error correction of speech recognition by custom phonetic alphabet input for ultra-small devices. In *Proceedings of the 2016 chi conference extended abstracts on human factors in computing systems* (p. 104–09).
- Gaida, C., P. Lange, R. Petrick, P. Proba, A. Malatawy, and D. Suendermann-Oeft (2014). *Comparing open-source speech recognition toolkits*. Retrieved from <http://antikenschlacht.de/su/pdf/oasis2014.pdf>

- Galibert, O. 2013. Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech. *Interspeech* 1131–34.
- Gerazov, B., and Z. Ivanovski (2013). A speaker independent small vocabulary automatic speech recognition system in macedonian. In *Proceedings of the second international conference taktos* (p. 5).
- Ghai, W., and N. Singh. 2012. Literature review on automatic speech recognition. *International Journal of Computer Applications* 41 (8):42–50.
- Glasser, A. 2019. Automatic speech recognition services: Deaf and hard-of hearing usability. *Extended Abstract* 1–6.
- Gong, Y. 1995. Speech recognition in noisy environments: A survey. *Speech Communication* 16 (3):261–91. doi:10.1016/0167-6393(94)00059-J.
- Guglani, J., and A. Mishra. 2018. Continuous punjabi speech recognition model based on kaldi asr toolkit. *International Journal of Speech Technology* 21 (2):211–16. doi:10.1007/s10772-018-9497-6.
- Haflan, V. (2019). *Noise robustness in small vocabulary speech recognition* (Master's thesis, NTNU). Retrieved from <http://hdl.handle.net/11250/2613396> (Master Thesis)
- Hannun, A. 2014. Deep speech: Scaling up end-to-end speech recognition. *ArXiv Prepr. ArXiv* 14125567.
- Hatala, Z. 2019. Practical speech recognition with htk. *ArXiv Prepr. ArXiv*190802119.
- Hauser, S., T. Sabir, and G. Thoma. (1999). Speech recognition for program control and data entry in a production environment. *Intelligent systems in design and manufacturingii*, vol. 3833, 24–34.
- Herbert, D., and B. Kang. 2019. Comparative analysis of intelligent personal agent performance. In *Pacific rim knowledge acquisition workshop*, 127–41.
- Hermansky, H., and N. Morgan. 1994. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing* 2 (4):578–89. doi:10.1109/89.326616.
- Hernández-Blanco, A., B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado. 2019. A systematic review of deep learning approaches to educational data mining. *Complexity* 2019:1–22. doi:10.1155/2019/1306039.
- Hibare, R., and A. Vibhute. 2014. Feature extraction techniques in speech processing: A survey. *International Journal of Computer Applications* 107 (5):1–8.
- Hinton, G., L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, et al. 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. doi:10.1109/MSP.2012.2205597.
- Hofe, R., S. R. Ell, M. J. Fagan, J. M. Gilbert, P. D. Green, R. K. Moore, and S. I. Rybchenko. 2013. Small-vocabulary speech recognition using a silent speech interface based on magnetic sensing. *Speech Communication* 55 (1):22–32. doi:10.1016/j.specom.2012.02.001.
- Huang, C., E. Chang, J. Zhou, and K.-F. Lee. 2000. Accent modeling based on pronunciation dictionary adaptation for large vocabulary mandarin speech recognition.
- Huang, F. 2011. An effective approach for Chinese speech recognition on small size of vocabulary. *Signal & Image Processing: An International Journal* 2 (2):48–60. doi:10.5121/sipij.2011.2205.
- Huggins-Daines, D., M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky (2006). Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 ieee international conference on acoustics speech and signal processing proceedings* (Vol. 1).
- Hui, J. (2019). Speech recognition — Asr decoding. *Medium*. Accessed June 09, 2021. Retrieved from <https://jonathan-hui.medium.com/speech-recognition-asr-decodingf152aebd779>

- Indurkha, N., and F. Damerou. 2010. *An overview of modern speech recognition xuedong huang and li deng*. In *Handbook of natural language processing*. 363–90. 2nd ed. Chapman and Hall/CRC.
- Ioffe, S. (2006). Probabilistic linear discriminant analysis. In *European conference on computer vision* (p. 531–42).
- Jelinek, F., R. Mercer, L. Bahl, and J. Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America* 62 (S1):63–63. doi:10.1121/1.2016299.
- Jia, Y. 2019. Direct speech-to-speech translation with a sequence-to-sequence model. *ArXiv Prepr. A rXiv* 190406037.
- Jiang, H. 2005. Confidence measures for speech recognition: A survey. *Speech Communication* 45 (4):455–70. doi:10.1016/j.specom.2004.12.004.
- Jinyu, L., D. Li, H.-U. Reinhold, and G. Yifan. 2016. Robust automatic speech recognition, 1–7. Elsevier.
- Johnson, S., P. Jourlin, G. Moore, K. Jones, and P. Woodland (1999). The Cambridge university spoken document retrieval system. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP 99* (Vol. 1, p. 49–52). Phoenix, AZ, USA.
- Jongman, S. R., Y. Khoe, and F. Hintz (2021). *Vocabulary size influences spontaneous speech in native language users: Validating the use of automatic speech recognition in individual differences research*. *Language and Speech* 64 (1). doi:10.1177/0023830920911079.
- Jordan, G., R. MacDonald, P. Jiang, J. Cattiau, R. Heywood, R. Cave, et al. (2021). *Automatic speech recognition of disordered speech: Personalized models outperforming human listeners on short phrases*. Retrieved from <https://doi.org/10.21437/Interspeech.2021-1384> (In Interspeech 2021, 4778–82. ISCA).
- Kala, T., and T. Shinozaki (2018). Reinforcement learning of speech recognition system based on policy gradient and hypothesis selection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 5759–63).
- Kaur, J., A. Singh, and V. Kadyan. 2021. Automatic speech recognition system for tonal languages: State-of-the-art survey. *Archives of Computational Methods in Engineering* (28):3.
- Képuska, V. 2017. Comparing speech recognition systems (microsoft api, google api and cmu sphinx). *International Journal of Engineering Research and Applications* 7 (3):20–24.
- Kraleva, R., and V. KraleV. 2009. On model architecture for a children’s speech recognition interactive dialog system. *Natural Sciences* 6.
- Krishnan, V., A. Jayakumar, and A. Babu. 2008. Speech recognition of isolated malayalam words using wavelet features and artificial neural network. *4th IEEE International Symposium on Electronic Design, Test and Applications* 240–43.
- Kurian, C. 2014. A survey on speech recognition in Indian languages. *(IJCSIT) International Journal of Computer Science and Information Technologies* (5):6169–75.
- Kurimo, M., M. Creutz, M. Varjokallio, E. Arsoy, and M. Saraclar. 2006. *Unsupervised segmentation of words into morphemes-morpho challenge 2005 application to automatic speech recognition*.
- Lee, K.-F. 1988. *Automatic speech recognition: The development of the sphinx system*, vol. 62. Springer Science and Business Media.
- Li, J. 2019. Jasper: An end-to-end convolutional neural acoustic model. *ArXiv Prepr. ArXiv* 190403288.
- Lima, T., and M. Costa-Abreu. 2020. A survey on automatic speech recognition systems for Portuguese language and its variations. *Computer Speech and Language* 62 (62):101055. doi:10.1016/j.csl.2019.101055.

- Liu, S., Y. Cao, N. Hu, D. Su, and H. Meng (2021). Fastsvc: Fast cross-domain singing voice conversion with feature-wise linear modulation. In *2021 IEEE International Conference on Multimedia and Expo (ICME)* (p. 1–6).
- Liu, X., and H. Zhou. 2014. A Chinese small vocabulary offline speech recognition system based on pocketsphinx in android platform. *Appl. Mech. Mater* (623):267–73.
- Liu, Y., R. Iyer, K. Kirchoff, and J. Bilmes. 2015. *Switchboardii and fisveri: High-quality limited-complexity corpora of conversational English speech*.
- Lokhande, R. (2015). *Feature extraction for asr: Intro— formula coding*. Retrieved from <http://wantee.github.io/2015/03/14/feature-extraction-for-asr-intro/> (accessed Jul. 01, 2021).
- Lucey, S., S. ran, and V. Chandran. 2001. Improving visual noise insensitivity in small vocabulary audio visual speech recognition applications. In *Proceedings of the sixth international symposium on signal processing and its applications (cat.no.01ex467)kuala lumpur*, vol. 2, 434–37. Malaysia.
- Maguolo, G., M. Paci, L. Nanni, and L. Bonan. 2019. *Audiogmenter: A matlab toolbox for audio data augmentation*. *ArXiv Prepr. ArXiv191205472*.
- Matarneh, R., S. Maksymova, V. Lyashenko, and N. Belova. 2017. Speech recognition systems: A comparative review. *Journal of Computer Engineering (IOSR-JCE)* 19 (5):71–79.
- Mayer, J. (2018). Low cost automatic speech recognition iot architecture for hands free assistance to people with mobility restrictions. In *Proceedings on the international conference on internet computing (icomp)* (p. 53–58).
- Mezzoudj, F., D. Langlois, D. Jouviet, and A. Benyettou. 2018. Textual data selection for language modelling in the scope of automatic speech recognition. *Procedia Computer Science* 128:55–64. doi:10.1016/j.procs.2018.03.008.
- Militaru, D., and M. Lazar (2017). Limited-vocabulary speaker independent continuous speech recognition using classbased language model. In *Proceedings of the annual symposium of the institute of solid mechanics and session of the commission of acoustics* (Vol. 58, p. 6).
- Minaee, S., A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang. 2020. Biometric recognition using deep learning: A survey. *ArXiv191200271 Cs. Apr.* 12. Retrieved from <http://arxiv.org/abs/1912.00271> [Online]. Available:)
- Mishaim, M., M. Malik, K. Mehmood, and I. Makhdoom. 2021. Automatic speech recognition: A survey. *Multimedia Tools and Applications* 80 (6):9411–57. doi:10.1007/s11042-020-10073-7.
- Mittal, P., and N. Singh. 2020. Subword analysis of small vocabulary and large vocabulary asr for punjabi language. *International Journal of Speech Technology* 23 (1):71–78. doi:10.1007/s10772-020-09673-3.
- Mnih, V., K. Kavukcuoglu, and D. Silver (2020). *Deep reinforcement learning projects*. Retrieved from <http://www.polodepensamento.com.br/blog/article.php?99f49d=deepreinforcement-learning-projects>. (accessed Aug. 27, 2020)
- Muhammad, G., Y. Alotaibi, and M. Huda. 2009. Automatic speech recognition for bangla digits. In *2009 12th international conference on computers and information technology*. Dhaka, Bangladesh: 379–83.
- Murphy, K. 2012. *Machine learning: A probabilistic perspective*. Cambridge, MA: MIT Press.
- Murshed, M., C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain. 2020. Machine learning at the network edge: A survey. *ArXiv190800080 Cs Stat* 15. Retrieved from <http://arxiv.org/abs/1908.00080> [Online]. Available.
- Myers, E. (2019). The role of artificial intelligence and machine learning in speech recognition. *Rev.* Retrieved from <https://www.rev.com/blog/artificialintelligence-machine-learning-speech-recognition>(accessed Feb. 06, 2020)

- Narang, S., and D. Gupta. 2015. Speech feature extraction techniques: A review. *International Journal of Computer Science and Mobile Computing* 4 (3):107–14.
- Novak, M. 2010. Evolution of the asr decoder design. In *Text, speech and dialogue*, ed. P. Sojka, A. Horak, I. Kopeček, and K. Pala., vol. 6231, 10–17. Berlin, Heidelberg: Springer.
- Noyes, J., and A. Starr. 2007. A comparison of speech input and touch screen for executing checklists in an avionics application. *The International Journal of Aviation Psychology* 17 (3):299–315. doi:10.1080/10508410701462761.
- O'Shaughnessy, D. 1988. Linear predictive coding. *IEEE Potentials* 7 (1):29–32. doi:10.1109/45.1890.
- Orosanu, L., and D. Jouvét. 2018. Adding new words into a language model using parameters of known words with similar behavior. *Procedia Computer Science* 128:18–24. doi:10.1016/j.procs.2018.03.003.
- Padmanabhan, J., and M. Premkumar. 2015. Machine learning in automatic speech recognition: A survey. *IETE Technical Review* 32 (4):240–51. doi:10.1080/02564602.2015.1010611.
- Park, D. 2019. *SpecAugment: A simple data augmentation method for automatic speech recognition*. *ArXiv Prepr. ArXiv190408779*.
- Petridis, S., J. Shen, D. Cetin, and M. Pantic (2018). Visual-only recognition of normal, whispered and silent speech. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (p. 6219–23).
- Pleshkova, S., A. BekyarSKI, and Z. Zahariev. 2019. Reduced database for voice commands recognition using cloud technologies, artificial intelligence and deep learning. In *2019 16th conference on electrical machines, drives and power systems*. Varna, Bulgaria (elma): 1–4.
- Pleva, M., J. Juhár, and A. Thiessen (2015). Automatic acoustic speech segmentation in praat using cloud based asr. In *2015 25th international conference radioelektronika (radioelektronika)* (p. 172–75).
- Povey, D. 2011. The kaldi speech recognition toolkit. *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.
- Prozorov, D., and A. Tatarinova. 2019. Comparison of grapheme-to-phoneme conversions for spoken document retrieval. In *2019 IEEE East-West Design and Test Symposium (EWDTS)*, 1–4.
- Qader, A., and H. Hassani (2019). Kurdish (sorani) speech to text: Presenting an experimental dataset. In *Arxiv191113087 cseess*. Retrieved from <http://arxiv.org/abs/1911.13087> (Accessed: Mar. 26, 2020. [Online]. Available.)
- Qiao, F., J. Sherwani, and R. Rosenfeld. 2010. *Small-vocabulary speech recognition for resource-scarce languages*. In *Proceedings of the first ACM symposium on computing for development - ACM DEV*, vol. 10, 1. London, United Kingdom.
- Rajapakse, T., S. Latif, R. Rana, S. Khalifa, and B. Schuller. 2020. Deep reinforcement learning with pre-training for time-efficient training of automatic speech recognition. *ArXiv Prepr. ArXiv200511172*.
- Rajashri, K., M. Ambewadikar, and M. Baheti. 2021. Review on small vocabulary automatic speech recognition system (asr) for marathi. *Open Access International Journal of Science and Engineering* 6.
- Rathinavelu, A., and G. Anupriya (2007). Speech recognition model for tamil stops. In *Proceedings of the world congress on engineering* (Vol. 1, p. 4).
- Rawat, S., P. Gupta, and P. Kumar. 2014. Digital life assistant using automated speech recognition. In *2014 innovative applications of computational intelligence on power, energy and controls with their impact on humanity*. (cipech) Ghaziabad, UP, India: 43–47.

- Renals, S., and G. Grefenstette. 2000. *Text-and speech-triggered information access: 8th elsnet summer school, Chios island*. revised lectures ed. Greece: Springer.
- Ronzhin, A., R. Yusupov, I. Li, and A. Leontieva. 2006. Survey of Russian speech recognition systems. *St Petersburg* 7:54–60.
- Saksamudre, S., P. Shrishrimal, and R. Deshmukh. 2015. A review on different approaches for speech recognition system. *International Journal of Computer Applications* 115:23–28.
- Salamon, J., and J. Bello. 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters* 24 (3):279–83. doi:10.1109/LSP.2017.2657381.
- Saon, G., and J.-T. Chien. 2012. Large-vocabulary continuous speech recognition systems: A look at some recent advances. *IEEE Signal Processing Magazine* 29 (6):18–33. doi:10.1109/MSP.2012.2197156.
- Schultz, T. 2009. Rapid language adaptation tools for multilingual speech processing. In *2009 IEEE workshop on automatic speech recognition understanding*, 51–51.
- Shangguan, Y., J. Li, Q. Liang, R. Alvarez, and I. McGraw. 2019. Optimizing speech recognition for the edge. *ArXiv Prepr. ArXiv190912408*.
- Shannon, C. 1951. Prediction and entropy of printed English. *Bell System Technical Journal* 30 (1):50–64. doi:10.1002/j.1538-7305.1951.tb01366.x.
- Sharma, R., S. Paladugu, K. Priya, and D. Gupta (2019). Speech recognition in kannada using htk and Julius: A comparative study. In *2019 international conference on communication and signal processing (iccsp)* (p. 0068–0072).
- Shi, H., and A. Maier. 1996. Speech enabled shopping application using Microsoft sapi. *International Journal of Computer Science Network Security* 6 (9):33.
- Sukumar, A., A. Shah, and P. Anto. 2010. Isolated question words recognition from speech queries by using artificial neural networks. In *2010 second international conference on computing, communication and networking*. Karur, India: 1–4.
- Supriya, S., and S. Handore. 2017. Speech recognition using htk toolkit for marathi language. In *2017 IEEE international conference on power, control, signals and instrumentation engineering*. Chennai: 1591–97.
- Tamgno, J., E. Barnard, C. Lishou, and M. Richomme (2012). Wolof speech recognition model of digits and limited-vocabulary based on hmm and toolkit. In *2012 uksim 14th international conference on computer modelling and simulation*, Cambridge United Kingdom (p. 389–95).
- Tejedor-García, C., V. Cardeñoso-Payo, and D. Escudero-Mancebo. 2021. Automatic speech recognition (asr) systems applied to pronunciation assessment of 12 Spanish for Japanese speakers, ” preprint. *PHYSICAL SCIENCES*. doi:10.20944/preprints202106.0687.v1.
- Thambiratnam, D., and S. Sridharan. 2000. Improving speech recognition accuracy for small vocabulary applications in adverse environments. *International Journal of Speech Technology* 9 (2):109–17. doi:10.1023/A:1009637129254.
- Vincent, E., T. Virtanen, and S. Gannot. 2018. *Audio source separation and speech enhancement*. Hoboken, NJ: John Wiley and Sons.
- Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* 13 (2):260–69. doi:10.1109/TIT.1967.1054010.
- Vu, N., T. Schlippe, F. Kraus, and T. Schultz. 2010. *Rapid bootstrapping of five Eastern European languages using the rapid language adaptation toolkit*.
- Warden, P. (2018). *Speech commands: A dataset for limited-vocabulary speech recognition*. Retrieved from <http://arxiv.org> (ArXiv180403209 Cs, Apr. 2018, Accessed: Jan. 02, 2020. [Online]. Available:)

- Whittaker, E., and R. Woodland (2001). Efficient class-based language modelling for very large vocabularies. In *2001 ieee international conference on acoustics, speech, and signal processing. proceedings (c at.no.01ch37221)* (Vol. 1, p. 545–48). Salt Lake City, UT, USA.
- Yadava, T., and H. Jayanna. 2017. A spoken query system for the agricultural commodity prices and weather information access in kannada language. *International Journal of Speech Technology* 20 (3):635–44. doi:[10.1007/s10772-017-9428-y](https://doi.org/10.1007/s10772-017-9428-y).
- Yi, L., Q. Li, B. Yang, Z. Yan, H. Tan, and Z. Chen. 2021. Improving speech recognition models with small samples for air traffic control systems,”. *Neurocomputing* 445(juillet):287–97. Retrieved from. doi:[10.1016/j.neucom.2020.08.092](https://doi.org/10.1016/j.neucom.2020.08.092).
- Young, S. 2002. The htk book. *Department of Engineering, University of Cambridge* 3 (175):12.
- Yu, S.-Z., and H. Kobayashi. 2003. An efficient forward-backward algorithm for an explicit-duration hidden Markov model. *IEEE Signal Processing Letters* 10 (1):11–14. doi:[10.1109/LSP.2002.806705](https://doi.org/10.1109/LSP.2002.806705).
- Zaykovskiy, D. 2006. Survey of the speech recognition techniques for mobile devices. *St Petersburg* 6.
- Zeng, D. 2015. Ai ethics: Science fiction meets technological reality. *IEEE Intell. Syst* 30 (3):2–5.
- Zhou, K., B. Sisman, R. Liu, and H. Li. 2021. *Emotional voice conversion: Theory, databases and esd. ArXiv Prepr. ArXiv210514762, 2021.*