IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 October 2022, accepted 24 November 2022, date of publication 2 December 2022,
date of current version 9 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3226539

RESEARCH ARTICLE

A Novel Human-Vehicle Interaction Assistive Device for Arab Drivers Using Speech Recognition

GHADEER A. JARADAT[1,2], MOHAMMAD A. ALZUBAIDI[2], (Member, IEEE), AND MWAFFAQ OTOOM[2], (Senior Member, IEEE)

[1]Electrical and Computer Engineering Department, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Computer Engineering Department, Yarmouk University, Irbid 21163, Jordan

Corresponding author: Mohammad A. Alzubaidi (maalzubaidi@yu.edu.jo)

This paragraph of the first footnote will contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456."

ABSTRACT About one-quarter of all car collisions in the United States are caused by distracted driving, and this ratio is expected to rise. As vehicles are equipped with more elaborate and complex technology, human-vehicle interaction via dashboard displays and controls will become more complex and distracting. Human-vehicle interaction via voice-based technology offers a less distracting alternative. In this study we aim to develop a voice-based car assistant, with a focus on Arabic language speech recognition. We prepare a new 4000-word domain-specific lexicon to comprehensively support driver-vehicle interactions, and we create corresponding text and speech corpora. Then we extract acoustic feature vectors and use various acoustic models to support speech recognition. The language model is created using an n-gram model. Then acoustic and language models, and the lexicon are combined to generate a decoding graph. The text corpus consists of 6110 elements, including words, phrases, and sentences. The speech corpus has more than 60000 recordings (almost 50 hours). For the decoding of noise-free audio, a Deep Neural Network + Hidden Markov Model provided 94.832% accuracy, a Subspace Gaussian Mixture Model + Hidden Markov Model provided 94.2% accuracy, and the best Gaussian Mixture Model + Hidden Markov Model provided 94.13% accuracy. For the decoding of noisy audio, a Deep Neural Network + Hidden Markov Model provided 93.316% accuracy, a Subspace Gaussian Mixture Model + Hidden Markov Model provided 92.62% accuracy, and the best Gaussian Mixture Model + Hidden Markov Model provided 91.82% accuracy. A usability study was conducted on the system with 10 participants. Almost all of the results of that study showed usability ratings of greater than 4.0 out of 5.0. These usability ratings indicate that the proposed system was seen by the participants as important, and useful for reducing driver distraction.

INDEX TERMS Arabic language, car assistant, human-vehicle interaction, speech recognition.

I. INTRODUCTION

About one-quarter of all car collisions in the United States are caused by inattentive or distracted driving. As wireless networking and entertainment systems become more widely employed in the auto industry, the number of distraction-related accidents is expected to rise [1]. According to research in [2], looking away from the road for more than two seconds increases the chance of a collision by four to twenty-four times.

Human-vehicle interaction using voice-based technology allows drivers to keep their eyes on the road while keeping their hands on the wheel, permitting drivers to more safely perform necessary tasks while driving [3]. Fig. 1 shows the sequence of events of a typical in-vehicle spoken dialog interaction.

Speech Recognition converts a user's voiced utterance into a textual hypothesis. That hypothesis is then parsed, and a semantic representation of the utterance is created, using Natural Language Understanding (NLU). Based on that semantic representation, a dialog manager produces a textual response (typically including prosodic markups) that will then be synthesized by a speech synthesizer.

Natural Language Processing (NLP) is a field within Artificial Intelligence in which linguistics is employed to help machines better understand, interpret, and generate

The associate editor coordinating the review of this manuscript and approving it for publication was Khursheed Aurangzeb.

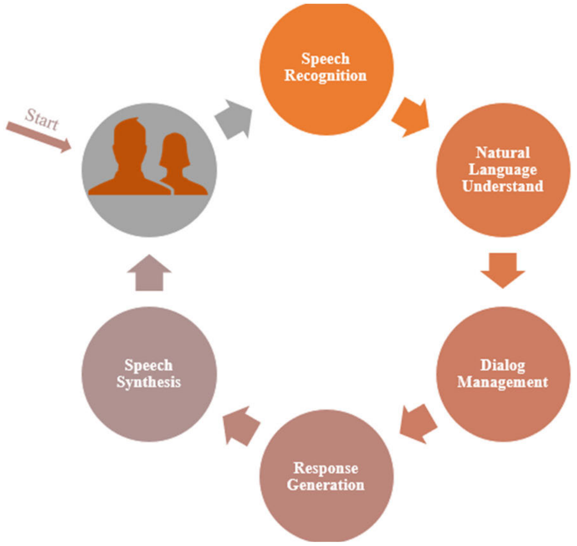


FIGURE 1. The sequence of events of a typical in-vehicle spoken dialog interaction, [3].

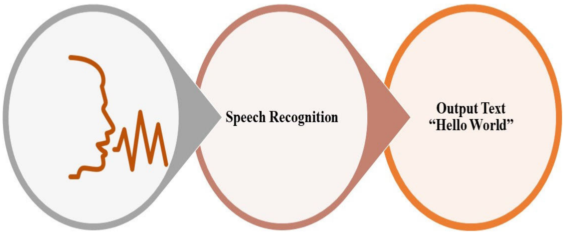


FIGURE 2. The speech recognition process, [7].

natural language [4]. Bridging the gap in oral communication between humans and computers is the main goal of Automatic Speech Recognition (ASR) [6]. It does so by transforming speech signals into computer readable texts, as shown in Fig. 2 [5].

Natural languages develop and evolve from everyday human use and repetition, without systematic building over time [4]. Natural languages include both spoken and signed languages [4]. According to some estimates, there are currently over 7000 human languages being used. [8].

Early research in ASR evolved from recognizing isolated digits spoken by a single person, to medium-sized continuous speech, and eventually progressed to large-vocabulary continuous speech recognition (LVSCR) [9]. ASR is currently being used in many applications, such as in mobile phones, automatic vehicles, industrial devices, military devices, and in many fields such as communication, education and medical and health care [10].

Arabic is one of the United Nations' six official languages [11]. It is the mother tongue of 206 million native speakers [12] and is listed as fifth, after Mandarin, Spanish, English and Hindi [4]. Modern Standard Arabic (MSA) is the official language used in the media, and is taught in schools and colleges within all Arab countries [13].

However, unlike ASR development for English, and for Asian languages such as Chinese and Mandarin, ASR for Arabic language has not yet been fully developed [14]. As a result, ASR for MSA has recently become the primary focus for a number of researchers.

Arabic ASR research is challenging, due to its lexical diversity and the scarcity of useful data [17]. One of the major problems facing Arabic ASR researchers is the shortage of written and spoken training data [12]. The most popular Arabic corpora are not available for free. They must be purchased from the Linguistic Data Consortium (LDC) or the European Language Resource Association (ELRA).

Arabic spoken corpora have been primarily gathered from radio and television news broadcasts and phone calls [12]. Because of the limitations of the available spoken corpora, Arabic ASR research and applications have been limited to particular domains, such as Arabic digits [15], [16], broadcast news [18], command and control [15], The Holy Qur'an [15], [23], and Arabic proverbs [19]. Limited text and speech Arabic corpora are also a major problem for Arabic ASR researchers who are seeking to apply Arabic ASR to a broader range of applications. This is of particular concern to researchers who would like to develop a versatile voice-based car assistant to facilitate human-vehicle interaction. Distracted driving accidents are expected to rise due to the increasing use of visually oriented human interfaces in cars. A voice-based car assistant could be useful for decreasing car accidents caused by distracted driving.

This study aims to develop a voice-based car assistant, with a particular focus on Arabic language speech recognition, to help in reducing driver distraction. Developing a lexicon, as well as speech and text corpora, is essential if we are to meet this objective.

This paper makes the following contributions:

- Building a domain-specific lexicon in Arabic to comprehensively support human-vehicle interaction with a dictionary of words in the domain, including a range of linguistic variations.
- Building speech and text Arabic corpora.
- Developing a complete ASR system, as a part of a car assistant system.

The remainder of this paper is structured as follows. Section II reviews speech production and perception, ASR methodologies and related work in the ASR area, and data sets available for this domain. Section III outlines the stages used in ASR including feature extraction, acoustic and language modeling, and decoding. It also introduces our evaluation metric and toolkit. Section IV illustrates the methodology used and the experimental setup for our ASR model, as well as our usability study. Section V presents and discusses the results of our model and of our usability study. Finally, Section VI provides the conclusion and discusses future work.

II. BACKGROUND AND LITERATURE REVIEW

In this section, a brief overview of sound generation, speech signal, and ASR approaches are given, a review for existing ASR models is given, and the available data sets are reviewed.

A. SPEECH PRODUCTION AND PERCEPTION

One of the essential communication channels between humans is speech and, to some degree, it is unique to every person.

Human speech is created from the vocal tract through the movement of many articulators such as the lips, the tongue, and the jaw. When people speak, air is expelled from the lungs via the trachea. This flow of air leads the vocal cords to vibrate, ultimately producing a variety of speech sounds. The resulting speech stream is received and processed by the human auditory system.

Speech communication can be divided into the following steps [20]:

- The speaker formulates his thoughts into phrases.
- The speaker generates a voice stream using the vocal cords and the speech system.
- That voice stream is conveyed acoustically through the air to the listener's ear.
- The resulting neural signals are transmitted via auditory nerves to the listener's brain.
- In the brain those neural signals are interpreted as language.
- The brain extracts meaning from the language interpretation.

Human speech has a frequency range of 85 Hz to 8 kHz, whereas human hearing has a frequency range of 20 Hz to 20 kHz. [4].

B. ASR METHODOLOGIES

ASR methodologies are classified into three approaches [20] as shown in Fig. 3: (1) the Acoustic Phonetic Approach (APA), (2) the Pattern Recognition Approach (PRA), and (3) the Artificial Intelligence Approach (AIA). All three of these approaches have one thing in common. They all depend heavily on feature extraction [21].

The Acoustic-Phonetic Approach treats a voice stream as a string of distinct phonetic units called phonemes - each with a distinctive set of acoustical features. This approach has not been widely used [22].

The Pattern Recognition Approach involves two crucial stages: pattern preparation and pattern comparison. The key advantage of this method is that it employs a mathematically defined training algorithm that can be trained with a series of labeled training samples, to ultimately create a useful representation of speech patterns [20]. This has become the predominant approach over the last six decades [23].

The Artificial Intelligence Approach combines the Acoustic-Phonetic Approach and Pattern Recognition Approaches [20], using Acoustic-Phonetic and Pattern Recognition theories and concepts. This approach has had only partial success [22].

C. DEVELOPMENT IN ASR MODELS

One of the earliest algorithms used for ASR is dynamic time warping (DTW), which is used to determine the optimal alignment between two-time series [24]. DTW has some drawbacks, including high complexity and the difficulty of comparing the elements from two separate voice streams. It is appropriate only for simple applications [24].

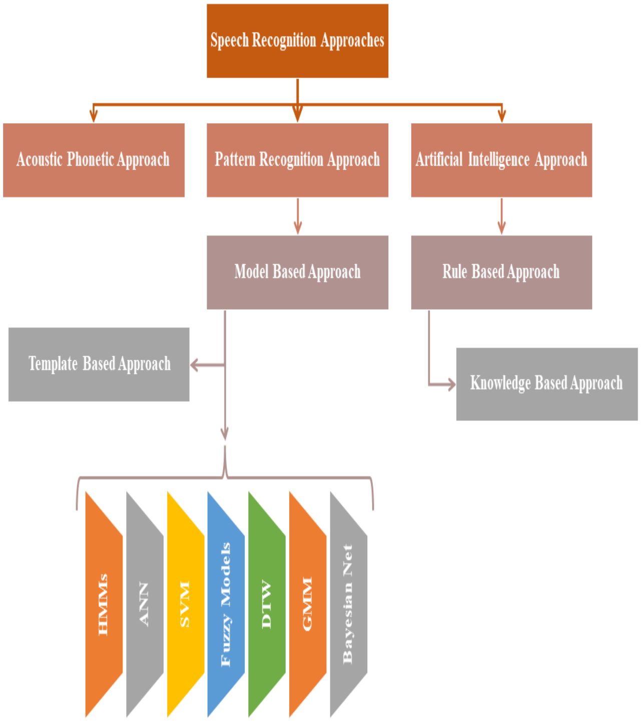


FIGURE 3. ASR methodologies classification, [20].

Due to limitations of DTW, ASR has moved to a statistical base - specifically to Hidden Markov Models (HMM) where a Gaussian Mixture Model (GMM) is trained to provide unique a probability density function for each HMM state.

Another mathematical approach to speech recognition, called Subspace Gaussian Mixture Model (SGMM), was proposed in [25], where all HMM states that employ the same number of Gaussians share the same GMM structure. This method was a modification on the GMM-HMM-based method, which involves training a distinct GMM for each HMM state.

SGMM has an advantage over GMM in that the number of parameters needed for each HMM state is small. This allows training with less data. It also allows the shared parameters to be trained using data from outside the domain and vocabulary. It was found that, when only a limited amount of training data (1 hour) was available, the SGMM-based model performed better than a conventional GMM-based model [26].

Research in [27] presented a Punjabi language speech recognition system for children's speech, which was developed using SGMM-HMM. It showed that using SGMM for acoustic modeling of small vocabulary datasets resulted in significant improvements for those small data sets. In particular, this research found that speech recognition based on SGMM outperformed GMM models for Punjabi children's speech.

Research has also shown that using Gaussian mixture models (GMMs) with HMM has a substantial drawback. They are statistically inefficient for modeling data on or near a nonlinear manifold in the data space [28].

A novel hybrid model architecture, called the Deep Neural Network-Hidden Markov Model (DNN-HMM), has been proposed, and has commonly been used in speech recognition in recent years. A deep neural network (DNN) is a form of neural network that is useful for detecting nonlinear relationships within data sets [28].

Several of these hybrid models have been presented, including shallow-NN-HMMs, and Multi-layer Perceptron HMMs (MLP-HMMs). Research in [28] presents comparative experiments for emotion recognition from speech. DNN-HMMs were compared with GMM-HMMs, shallow-NN-HMMs, and MLP-HMMs. The results showed that (when the hidden layers and hidden unit numbers were appropriately configured) DNN-HMM provides better labeling than GMM-HMM. Overall, the DNN-HMM model produced the best performance of all these models.

In the 1990s, machine learning was incorporated into ASR, resulting in increased accuracy [4]. Researchers in [29] introduced a concept of an end-to-end ASR system, using only recurrent neural networks (RNN) - which is a class of deep neural network (DNN) - instead of combining GMM with HMM, or DNN with HMM. However, a significant volume of training data is needed to train RNNs. To provide adequate training data, an extensive data set (consisting of 5000 hours of reading speech in English) was collected. The trained RNN model provided a 9.2% Word Error Rate (WER) for the raw data, and a 9.0% WER for the same 5000 hours plus noise.

The Time Delay Neural Network (TDNN) has been shown to be an effective network system for ASR, due to its ability to model context. Furthermore, TDNN is faster to train than RNN because it is a feed-forward neural architecture [30]. In [31], researchers compared TDNN, a Convolutional Neural Network (CNN), DNN, and HMM-GMM on Myanmar language. The experiment was done using the Kaldi toolkit on 76 hours and 53 minutes of training data. The WER for TDNN was 15.03%. That significantly outperformed CNN (18.44% WER), DNN (20.20% WER) and HMM-GMM (26.11% WER).

D. ARABIC ASR

As mentioned before, a number of researchers have been focusing on developing ASR for the Arabic language. Researchers in [32] used a DTW-based system for recognition of Arabic digits, resulting in a recognition accuracy of 77%.

Research in [15] developed 3 corpora; namely the Holy Qur'an Corpus (HQC-1) around 18.5 hours, the command-and-control corpus (CAC-1) around 1.5 hours, which is labeled as a small vocabulary set (around 30 words in the lexicon), and the Arabic digits corpus (ADC) less than one hour of speech. They used Mel Frequency Cepstral Coefficient (MFCC) to perform feature extraction, a Gaussian Mixture Model (GMM) for generating probability density functions, and Carnegie Mellon University's CMUSphinx-IV engine, which is based on Hidden Markov Models (HMMs). This produced a word recognition rate of 99.21% for the digits corpus, 98.1% for the command-and-control corpus, and 70.8% for the Holy Qur'an corpus.

Research in [16] developed a corpus for Arabic digits, and used CMUSphinx-IV for voice recognition based on HMM, with a recognition rate of 85.56% for male speakers and 83.34% for female speakers.

In [17] the authors used Qatar's lexicon which has 526K distinct words, with 2M different pronunciations to build three Arabic broadcast news speech recognition systems. The DNN-HMM WER was 29.81%. This outperformed SGMM-HMM (32.94% WER) and GMM-HMM (36.74% WER).

Research in [33] presented an end-to-end multi-dialectal Arabic language speech recognition system, where a huge multi-dialectal Arabic speech corpus (consisting of approximately 1,400 hours of speech) was developed and used to train a Convolutional Neural Network + Recurrent Neural Network (CNN-RNN) model, resulting in an overall 14% WER.

E. AUTOMOBILE DATA SET

Vehicles fitted with both person and automobile sensors have been deployed during the last two decades to gather realistic data on drivers, vehicles, and driving conditions. Speech corpora have been obtained from in-vehicle usage to improve in-vehicle Automatic Speech Recognition (ASR) and spoken dialog systems [3].

SPEECHDAT-CAR was the first international program to provide a multilingual speech corpus for automobile applications, with ten languages (American English, British English, Danish, Finnish, French, Flemish/Dutch, German, Greek, Italian and Spanish) [34].

Research in [35] formulated and analyzed a new acoustic speech corpus for creating in-vehicle interactive navigation and route planning systems. Data was gathered from over 1000 speakers from all over the United States.

F. ARABIC DATA SETS

Several Arabic lexicons have been created for various purposes. Qatar's lexicon is very large (2,022,708 words). However, it has certain linguistic issues that are not systematic. Other available lexicons, such as the CALLHOME and the Madar lexicons are in particular Arabic dialects. Some available lexicons, such as the Buckwalter Arabic Morphological Analyzer Version 1.0 do not contain Arabic diacritics. None of the available lexicons are domain specific, covering narrow topics such as numbers, or broad topics such as Qur'an, news, or a variety of other non-specific topics.

G. RESEARCH GAP

Across the speech recognition work to date, comparatively little research has been done in Arabic speech recognition, and the existing work has been limited to particular research topics, such as Qur'an, digits, and news. In particular, there has been no work aimed at supporting an Arabic car assistant. Because they do not include the words or sentences that are needed for such an application, we could not use any of the existing lexicons or corpora to support our development of a car assistant because they are not domain specific and have many certain linguistic issues that are not systematic, many are for dialects not for standard Arabic, some do not contain Arabic diacritics. With this in mind, we propose our initial research question:

Research Question 1: How could we develop an automatic speech recognition system for an Arabic car assistant?

The literature shows that the limitation of Arabic ASR comes from the lack of adequate lexicons, as well as training and testing data. While several Arabic lexicons have been produced for a variety of purposes, none of them were specifically designed for human-vehicle interaction. This brings us to our second research question:

Research Question 2: How can we develop a lexicon, as well as text and speech corpora in Arabic specifically for the car assistant domain?

III. CONCEPTUAL FRAMEWORK

This section provides a detailed description of the architecture of a typical speech recognition system, including feature extraction and decoding methods.

A. ARCHITECTURE OF AN ASR SYSTEM

ASR involves the study of speech signals, and the methods to interpret these signals into words. To generate speech signals, people use their vocal cords. The resulting sound is captured using a high-quality microphone and streamed through a speech recognition device that interprets and converts the signal stream into a sequence of words (text).

Fig. 4 shows the consecutive steps involved in the speech recognition process [5]: Pre-processing, Feature Extraction, Decoding (the actual speech recognition) and Post-Processing to produce a text string.

Various methods and algorithms are used for each of these steps to build an ASR system. The ones that we used for each step are detailed later.

The 4-step ASR process detailed in Fig. 4 can be reduced to two basic stages [36]:

A- The Acoustic Front-End stage converts the audio speech stream into a stream of feature vectors in digital machine format.

B- The Decoding and Post-Processing stage is trained to identify likely matches between the incoming string of feature vectors and words, thus producing a list of plausible word sequences (i.e. N-best hypotheses) and Post-Processing then selects from among these hypotheses.

A typical speech recognition system is designed with major components that involve the acoustic front-end, acoustic model, lexicon, language model and decoder, as seen in Fig. 5 [5], [23].

The Acoustic Front-End translates the continuous speech signal into a string of discrete feature vectors. During Pre-Processing, the speech signal is enhanced, by applying pre-emphasis filters plus noise removal or reduction [7]. Feature Extraction then converts the resulting audio signal into a string of fixed-size acoustic feature vectors [7]. The Decoding process then applies acoustic and language models to identify the most likely matches (i.e. the most likely hypotheses) between the incoming string of feature vectors and all of its internally stored sequences of words [7]. Post-Processing then selects the most likely hypothesis from among the n-best hypotheses [7], [23].

1) MATHEMATICAL REPRESENTATION OF ASR

Statistical ASR determines the most likely word sequence, given a speech signal, which might come from a real-time audio input stream, or from an audio recording [4]. The aim is to identify the most likely word sequence w^* given a sequence of acoustic feature vectors X [4].

$$w^* = arg_w \ max \left\{ P\left(w|X\right) \right\}$$

using Bayes' Rule, the expression is transformed to

$$w^* = arg_w \ max \left\{ \frac{P\left(X|w\right).P\left(w\right)}{P\left(X\right)} \right\}$$

Since $P\left(X\right)$ is independent of the sequence of words, it is typically deleted from the expression

$$w^* = arg_w \ max \left\{ P\left(X|w\right).P\left(w\right) \right\}$$

where $P\left(X|w\right)$ is the Decoded likelihood of a sequence of feature vectors conditioned by some sequence of words w, as determined by the Acoustic and Language Model, and $P\left(w\right)$ is what will be determined by the Post Processing language model.

2) PRE-PROCESSING

In ASR systems, Pre-Processing is the first step of speech recognition; it adjusts or modifies the speech signal so that it will be more suited for the Feature Extraction process [37]. The main factor to consider in speech signal pre-processing, is whether the signal is corrupted by some background or ambient noise.

During this Pre-Processing stage, several enhancement techniques and operations can also be performed. One of the first enhancement techniques that could be applied is pre-emphasis filtering. Its purpose is to compensate for lip radiation and the high-frequency attenuation that occurs during the sampling process. High-frequency components are emphasized, and low-frequency components are deemphasized [38].

Other techniques that might be used are speech enhancement techniques, which are aimed at channel and noise compensation in adverse environments [38].

3) FEATURE EXTRACTION

Effective feature extraction is vital in ASR systems [39]. It recognizes and distinguishes one speech feature from

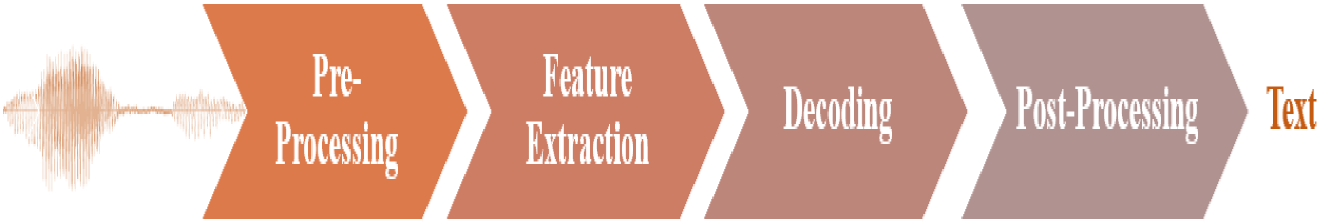


FIGURE 4. ASR process steps.

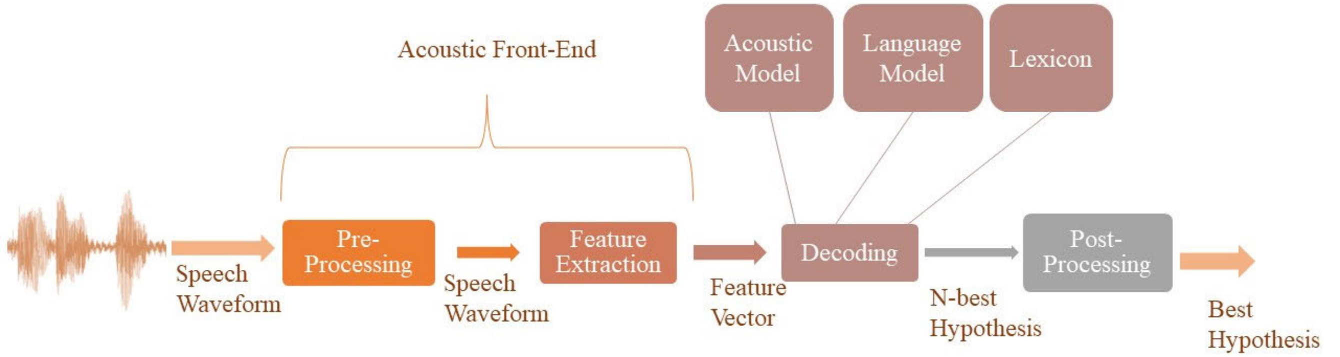


FIGURE 5. Architecture of typical speech recognition system, [5].

another [40]. The aim of this stage is to extract the most useful feature vectors for recognition [7].

Various methods are used for feature vector extraction, such as Linear Prediction Coding (LPC), Mel Frequency Cepstral Coefficient (MFCC), Linear Discriminant Analysis (LDA), Perceptual Linear Prediction (PLP), Principal Component Analysis (PCA), Cepstral Mean and Variance Normalization (CMVN) and more [39].

The best known and most widely used feature extraction technique for speech recognition is MFCC [40]. It is described in detail here, as it is used in this work.

The Mel Frequency Cepstral Coefficient (MFCC) algorithm, (when applied to a short segment of audio extracted from a voice stream) produces a set of coefficients that represent the frequency response of the speaker's vocal tract to glottal pulses during that segment. It is a very fast, reliable and easy computational technique [41]. Several steps are used to compute the MFCC feature vectors. Fig. 6 shows these steps.

1) Frame Blocking: The frequency response of the human vocal tract to glottal pulses can change rapidly during speech [42] This is the reason feature vectors are extracted from short segments of the speech signal. The speech signal is chopped into short "frames" (usually between 5 and 100 milliseconds long) with a 10 ms overlap with the previous and subsequent frame. This results in a resolution of 10 ms for frames [4]. The purpose of this overlapping scheme is to smooth the transition from frame to frame [43].
2) Windowing: This modulates the amplitude of the samples across each frame with a scaling function. The aim of this process is to remove discontinuities at the edges of frames [42]. Hann window and Hamming window are the most commonly used in ASR [4].
3) Fast Fourier Transform (FFT): This computes the Discrete Fourier Transform (DFT) across each frame. It provides a spectral profile of the frequency response of the vocal tract within that frame. [7].
4) Mel Filter Bank: A Mel filter bank applies a set of triangular bandpass filters to the FFT power spectrum to extract a coefficient for each frequency band. These filters mimic the non-linear perception of frequencies by the human ear, which is linear up to 1 kHz, and then logarithmic above that [42]. The filters within the Mel filter bank are logarithmic at higher frequencies and linear at lower frequencies [4]
5) Cepstrum: This maps the amplitude coefficients produced by the Mel Filter Bank (which collectively represent the spectral frequency response of the vocal tract during the frame) back into the time domain, using the Discrete Cosine Transform (DCT). (The DCT function is used instead of the inverse Fourier Transform because it is more efficient and produces real-number coefficients instead of complex numbers.) These real-number cepstral coefficients are then used to create a feature vector for that frame [4], [42].

The temporal first and second derivatives of cepstral coefficients from frame to frame provide additional information about the temporal changes in the vocal tract. The first-order derivatives (called delta coefficients Δ) indicate the rate at which the vocal tract is changing, which is an indicator of the rate speech, while the second-order derivatives (called delta-delta coefficients $\Delta\Delta$) represent changes in the rate of speech.

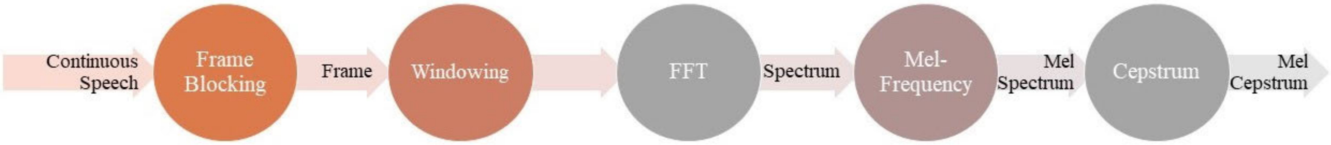


FIGURE 6. MFCC features extraction steps.

4) DECODING

As mentioned earlier, the decoding stage (i.e. the Acoustic Model and the Language model) tries to find candidate sequences of words (w^*) that are most likely to match the sequence of incoming feature vectors (X), where the probability value for each word $P(X|w)$ is determined by the Acoustic Model and the value of the $P(w)$ is determined by the Language Model.$***$

5) ACOUSTIC MODELING

Acoustic modeling computes a likelihood that a series of feature vectors extracted from the voice stream matches each word in the lexicon. In other words it estimates the likelihood $P(X|w)$. The likelihood for individual words is computed by concatenating the likelihood of simple sub-word components (called phones) based on a pronunciation lexicon [6]. Note: A speech recognizer can match spoken words that did not exist in its training set to words in its internally stored lexicon by recognizing sub-word units (i.e. phones) that were learned from the words that were in its training set.

A word is composed of a sequence of phones. In large vocabulary speech recognition systems (more than 5000 words), training might be based on monophones (i.e. single phones), on diphones, (two sequential phones), or triphones (three sequential phones).

The acoustic model might be built using a variety of approaches, including a Hidden Markov Model + Gaussian Mixture Model (HMM-GMM), a Hidden Markov Model + Subspace Gaussian Mixture Model (HMM-SGMM), a Hidden Markov Model + Deep Neural Network (HMM-DNN), conditional random fields, segmental models, and maximum entropy models. The Hidden Markov Model is used extensively in speech recognition, and is considered one of the best statistical models.

6) LANGUAGE MODELING

The Language Model $P(w)$ receives strings of phones from the Acoustic model and determines which strings of phones are valid words in the language, and in what order those words can appear [23]. In doing so, it computes a probability for a given series of words. To put it another way, if there is a sequence of words, $w = (w_1, w_2, \ldots, w_k)$, the language model computes the probability $P(w_1, w_2, \ldots, w_k)$ of that particular sequence occurring in the language.

There are many types of Language Modelling techniques used in ASR. However, the n-gram model is the most common. The n-gram model uses prior words to estimate the likelihood of a subsequent word, as described below [23]:

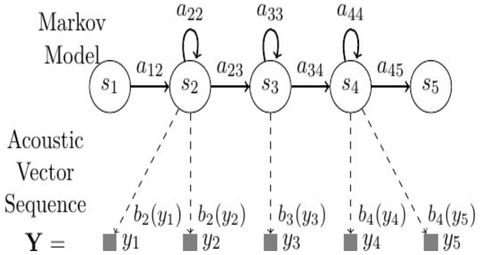


FIGURE 7. HMM-based phone model, [47].

$$P(w) = \prod_{k=1}^{k} P(w_k | w_{k-1}, w_{k-2}, w_{k-3}, \ldots, w_1)$$

This process is based on the premise that the probability of each word depends on the previous words. The bigram and trigram are forms of n-gram language modelling that are commonly applied in ASR. The bigram form is when n of the n-gram equals 2 (i.e. one prior word) and the trigram when n equals 3 (i.e. two prior words).

7) POST-PROCESSING

Most Decoders generate a list of plausible word strings (i.e. hypotheses) sorted by their statistical likelihood. Since this is a list of the best n hypotheses, it is known as the n-best list. The Language Model in the Decoder scores each of the n-best hypotheses based on their plausibility. Post-processing then chooses from among this list of word sequence candidates. The hypotheses that it assigns the highest score is then used for recognition [5].

Additional Post-Processing algorithms might recover punctuation, add capitalization, and use abbreviations. In addition, numbers and other forms of special data can be translated from words to regular form. This is all done to increase readability.

B. HIDDEN MARKOV MODEL (HMM)

The Hidden Markov Model (HMM) is one of the most popular statistical modeling techniques. It was presented and researched during the 1960s and 1970s. The HMM is an augmentation of the Markov chain [44], [45], [46].

The implementation of an Acoustic Model as a Hidden Markov Model is shown in Fig. 7 [47].

It was mentioned earlier that the incoming speech stream is chopped into a string of short frames. Because of the shortness of those frames, a single phone might span several frames.

Fig. 7 shows one possible sequence of state changes within an Acoustic Model's Hidden Markov Model. The extraction of a sequence of feature vectors from the voice stream is represented by y_1, y_2, y_3, y_4 and y_5. States s_2, s_3, and s_4 are states that represent phone A, phone B and phone C, respectively.

There are three transition possibilities for each of states s_2, s_3 and s_4. For example, a_{12}, a_{22} and a_{23} represent the three state transitions for s_2, When in State s_1, the probability of a transition a_{12} into a state s_2 (which represents phone A) is greater if a feature vector resembling phone A is extracted from the voice stream. If subsequent feature vectors also resemble phone A, the ''looping'' transition a_{22} is likely. However, if a feature vector resembling phone B is extracted, a transition out of state s_2 to state s_3 (which represents phone B) is likely. Thus, a transition into a state, followed by a transition out of that same state indicates the passing of a single phone in the voice stream, from start to finish [48].

As the ongoing voice stream is processed, if the sequence of extracted feature vectors (y_1, y_2, y_3, y_4, y_5) creates a high probability of transitions s_1 s_2 s_3 s_4 s_5, then it is highly probable that the sequence of phones represented by states s_2, s_3, and s_4. (i.e. phone A, phone B, phone C) was present in the voice stream.

If there is ambiguity in some of the extracted feature vectors, there might be significant probabilities for state transitions into several different states. When this happens, the Hidden Markov Model might produce significant probabilities for more than one sequence of phones.

The degree of similarity between each feature vector and all the possible phones is determined by a classifier, which assigns a similarity value between each feature vector and each of the possible phones. Such a classifier might be implemented with a Gaussian Mixture model (GMM), a Subspace Gaussian Mixture Model (SGMM), or a Deep Neural Network (DNN).

C. GAUSSIAN MIXTURE MODEL (GMM)

A Gaussian Mixture Model (GMM) is a classification algorithm that employs weighted sums of Gaussian densities to collectively represent irregular regions within a vector space [49].

In the case of Acoustic Modeling, each of these regions would represent a particular class of feature vectors (i.e. a particular phone). In some cases, the regions representing different phones might overlap. In that case a feature vector in the overlap region would be assigned probabilities for membership in both of the overlapping classes. This is called statistical classification. GMM classifiers are widely recognized for their ability to statistically classify vectors within vector spaces where classes are represented by irregularly shaped and overlapped regions, and they have been found to be useful for a wide range of applications [56].

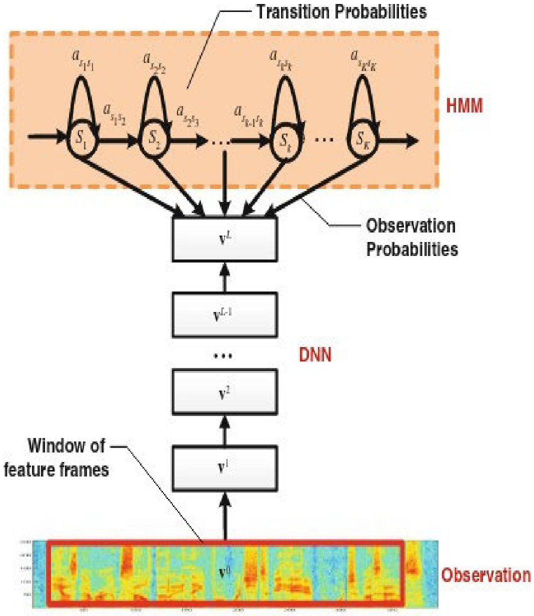


FIGURE 8. DNN-HMM hybrid system architecture, [51].

D. SUBSPACE GAUSSIAN MIXTURE MODEL

In the Subspace Gaussian Mixture Model (SGMM) all classification regions in the vector space are defined by the same Gaussian Mixture Model structure, with the same number of Gaussians used to define each region [25]. The model is defined by state vectors and a global mapping from vector space to GMM parameter space [25]. Compared to a standard GMM model, this model appears to produce better classification results [25].

E. DEEP NEURAL NETWORK

An alternative to classification using Gaussians to define regions within a Feature Vector space is to use a Deep Neural Network (DNN) [50]. A DNN is a feed-forward artificial neural network with more than one hidden layer between its input and output layers. DNNs with multiple hidden layers outperform GMM classification on a number of speech recognition criteria - sometimes by a significant margin [50]. Through the use of multiple hidden layers, a deep neural network can provide classification of feature vectors for very complex data sets.

For a DNN-HMM hybrid system, the HMM represents the speech signal's sequential characteristic, whereas the DNN provides the statistical classification of the feature vectors that drive the HMM. Fig. 8 shows the architecture of the DNN-HMM hybrid system [51].

F. TRIPHONE STATE TYING

The beads-on-a-string model, which represents all spoken utterances by concatenating a sequence of context-independent phones (aka monophones) has a key flaw in that it fails to reflect the high degree of context-dependent variation in phones during real speech [52]. To overcome this problem, we can use a Triphones model, where a different phone model is selected for each possible combination of its neighbors [52].

However, using a different phone model for each possible combination of two neighbors greatly increases the number of models that must be created during training. For example, if there are N phones, there will be N^3 triphones. Fortunately, it is not necessary to model all possible triphones, because not all phones vary significantly in different contexts. State tying is used to employ the same phone model in multiple triphone models, thus simplifying the modeling process [47], [52], [61].

G. DISCRIMINATIVE TRAINING

Discriminative training establishes an objective function for differentiating between possible hypotheses [54]. Different forms of objective functions are used in speech recognition systems. Those used in this study are described below:

1) Maximum Mutual Information (MMI): The goal of this function is to maximize the posterior probability of the correct word sequence. This model is given by [55]:

$$\mathcal{F}_{MMI}(\lambda) = \sum_{r=1}^{R} log \frac{\mathcal{P}_\lambda \left(\frac{X_r}{M_{sr}}\right)^k P(s_r)}{\sum_s \mathcal{P}_\lambda \left(\frac{X_r}{M_s}\right)^k P(s)}$$

λ represents the parameters of the acoustic model, X_r denotes to the training words, M_s represents the HMM sequence corresponding to a sentence s, and s_r is the correct transcription for the $r'th$ utterance, k is the acoustic scale as used in decoding and $P(s)$ is the language model.

2) Minimum Phone Error (MPE): This seeks to decrease phone error rates in order to improve phone level accuracy. MPE is defined as follows [55]:

$$\mathcal{F}_{MPE}(\lambda) = \sum_{r=1}^{R} \frac{\sum_s \mathcal{P}_\lambda \left(\frac{X_r}{M_{sr}}\right)^k P(s) A(s, s_r)}{\sum_s \mathcal{P}_\lambda \left(\frac{X_r}{M_s}\right)^k P(s)}$$

where $A(s, s_r)$ denotes to the raw phone accuracy of s when compared to the reference s_r, which equals the number of accurate phones minus the number of insertions, k denotes to the acoustic model scaling factor.

H. ADAPTATION TECHNIQUES

In most statistical modeling the model's training and input circumstances aren't always the same. Speaker variances, background noise, and channel differences can lead to poor recognition performance [56]. Acoustic model adaptation changes the parameters of an acoustic model used for speech recognition to better match the actual acoustic features [56].

Speaker adaptation is the process of employing a mapping function f from the space of parameters of the initial models to the space of the goal model [56].

$$\hat{\theta} = f(\theta_1 \ldots \theta_n)$$

where $\hat{\theta}$ represents the target model that must be obtained, f represents the adaptation model, and n is the number of initial models provided.

Maximum-Likelihood Linear Regression (MLLR) is an adaptation technique that is widely used in the ASR field. This method adapts to a given speaker by improving the probability between the real model and the adaptation model, which is achieved by linear modification of the Gaussian model parameters [57].

There are several variations on MLLR, such as mean only-MLLR, standard MLLR, and feature space MLLR (fMLLR), also known as constrained MLLR. fMLLR is calculated by performing linear transformations on the observation characteristics rather than on the model parameters [57].

I. EVALUATION

Word Error Rate (WER) is the most widely used measure of speech recognition performance. Using the Levenshtein distance measure, WER calculates the edit distance between the prediction and the target, based on the required number of insertions, deletions, and substitutions [4]. WER is defined as:

$$WER = \frac{Insertion\,(I) + Substitution\,(S) + Deletion\,(D)}{Total\ Number\ of\ Reference\ Words\,(N)}$$

J. KALDI TOOLKIT

Kaldi is an open-source speech recognition toolkit written in C++ and distributed under the Apache License v2.0. Kaldi's goal is to provide a code that is modern, flexible, easy to understand, modify and extend [58]. Other toolkits such as HTK, CMUSphinx and the RWTH toolkit are available. However, Kaldi was chosen due to its features: its code is integrated with Finite State Transducer (FST), it has extensive linear algebra support, it has an extensible design where algorithms are provided in the most generic form, it has a free license, it provides complete recipes for creating ASR systems, and its code has been tested extensively to ensure that it returns high accuracy results [58].

1) FINITE STATE TRANSDUCER

The Finite State Transducer (FST) is a finite state automaton that labels its states with input and output symbols and converts between input and output sequences. If the FST is labeled with inputs, outputs and weights, then it is called Weighted Finite State Transducer (WFST), where these weights can be used to indicate a duration, a probability, or a cost [59].

In general, Kaldi employs WFST in almost all training and decoding algorithms to merge acoustic and the language model information. For speech recognition the four main models are word-level grammar G, the pronunciation lexicon L, the context-dependency transducer C, and the HMM transducer H [60].

In Kaldi, the decoding graph may be constructed by creating the HCLG graph, which is described below using the associative operation:

$$H\ o\ C\ o\ L\ o\ G$$

2) WORD LATTICE

The creation of word lattices is an effective solution for dealing with speech decoding's big dimensional search challenge. A word lattice represents the various word-sequences that are ''sufficiently probable'' for a given speech stream [47].

The task of generating the transcription for a sequence of feature vectors by computing the Mel Frequency Cepstral Coefficients (MFCCs) from the audio input is equivalent to finding the most likely path through the Weighted Finite State Transducer (WFST), i.e. the path that has the lowest final cost.

For an utterance with T frames the search graph is defined as:

$$S = U\ o H\ C\ L\ G$$

where U is the Weighted Finite State Acceptor (WFSA).

An acceptor is a Weighted Finite State Transducer with identical input and output symbols, corresponding to the utterance. It has $T + 1$ states, with an arc for each combination of (time, context-dependent HMM state). The associated acoustic probability is the cost of each arc. Finding the best path across S is the decoding problem's equivalent [61].

Viterbi decoding with beam-pruning is a common search technique used in this stage. It can be summarized as follow [62]:

- All parallel arcs are directly compared for each time frame. Paths with probability massively lower than the most likely path are eliminated. Assume that α is the beam parameter. If the most probable path has the score p_{MAX}, paths with individual scores p_i that meet the criteria $|p_i - p_{MAX}| > \alpha$ will be trimmed.

A word lattice is then constructed from the paths that have survived, and that meet some additional requirements.

Finding the most likely path across the word lattice is a search problem that can be solved with any rapid dynamic search technique. The Viterbi Algorithm is a common choice, and the one used by Kaldi.

IV. METHODOLOGY

A. OVERVIEW

First, we prepared the data set (lexicon, text corpus, and speech corpus with its transcriptions). Then pre-emphasis filtering was used as a pre-processing step. Then we used the MFCC algorithm to extract features with 25 ms frames, shifted by 10 ms each time, with a Hamming windowing function. GMM, SGMM and DNN were then used to classify the feature vectors and estimate the transition probabilities within the HMM model.

The language model was created using an n-gram model and represented using WFST. Then, a decoding graph was created to combine the HMM structure with the lexicon and language model, in the form of WFST. For test data, after the pre-processing step and extraction of the MFCC features and decode using the decoding graph, a lattice of probable sequences of words was generated and used for scoring. Fig. 9 shows the overall methodology used in our study.

B. DATA SET

It was necessary to create a new domain-specific lexicon, as well as text and speech corpora. This domain specific lexicon was built to cover the requirements of a car assistant, with a comprehensive dictionary of words in that domain, and with their possible linguistic variations using the International Phonetic Alphabet (IPA) for Arabic language.

1) EXPERIMENT 1: PREPARING THE LEXICON

The lexicon should include all words, in Modern Standard Arabic (MSA), relevant to the human-vehicle interaction domain (as well as their various linguistic variants) transcribed using the International Phonetic Alphabet (IPA). The IPA is a collection of symbols used to represent the speech sounds of many languages throughout the world [63]. The lexicon was built in Experiment 1, which was conducted using the following steps:

1) Ten car drivers provided words to build the lexicon. The drivers were then asked about the ways in which they might interact with a car using this lexicon.
2) We consulted with an expert linguist to list all the words related to the ways in which drivers might interact with a car.
3) The expert linguist provided a comprehensive list of possible variations for the words in the lexicon.
4) The expert linguist transcribed each of the words in the lexicon, using IPA.

2) EXPERIMENT 2: GENERATING THE CORPUS

We needed to create text and speech corpora. A text corpus is a collection of texts (or portions of texts) that can be subjected to generic linguistic analysis [64]. A speech corpus is a collection of audio recordings, with spoken words/sentences and text transcriptions [65].

With the assistance of the expert linguist we prepared the text corpus - a set of relevant phrases and sentences that employed the words in the lexicon.

To prepare the speech corpus, we conducted Experiment 2, using the following steps:

1) We divided the speech corpus up into small sections.
2) 46 car driver (19 males, 27 females) with ages ranging from 18 to 75 were asked to record phrases and sentences from the speech corpus, with the technical specifications shown in Table 1.
3) Copies of the audio recordings were blended with background traffic noise.
4) Python scripts were developed to transcribe the recordings of the uttered sentences, phrases or words.

C. LANGUAGE MODEL

We used the SRILM toolkit to build a bigram language model. SRILM is a set of C++ libraries, executable programs, and

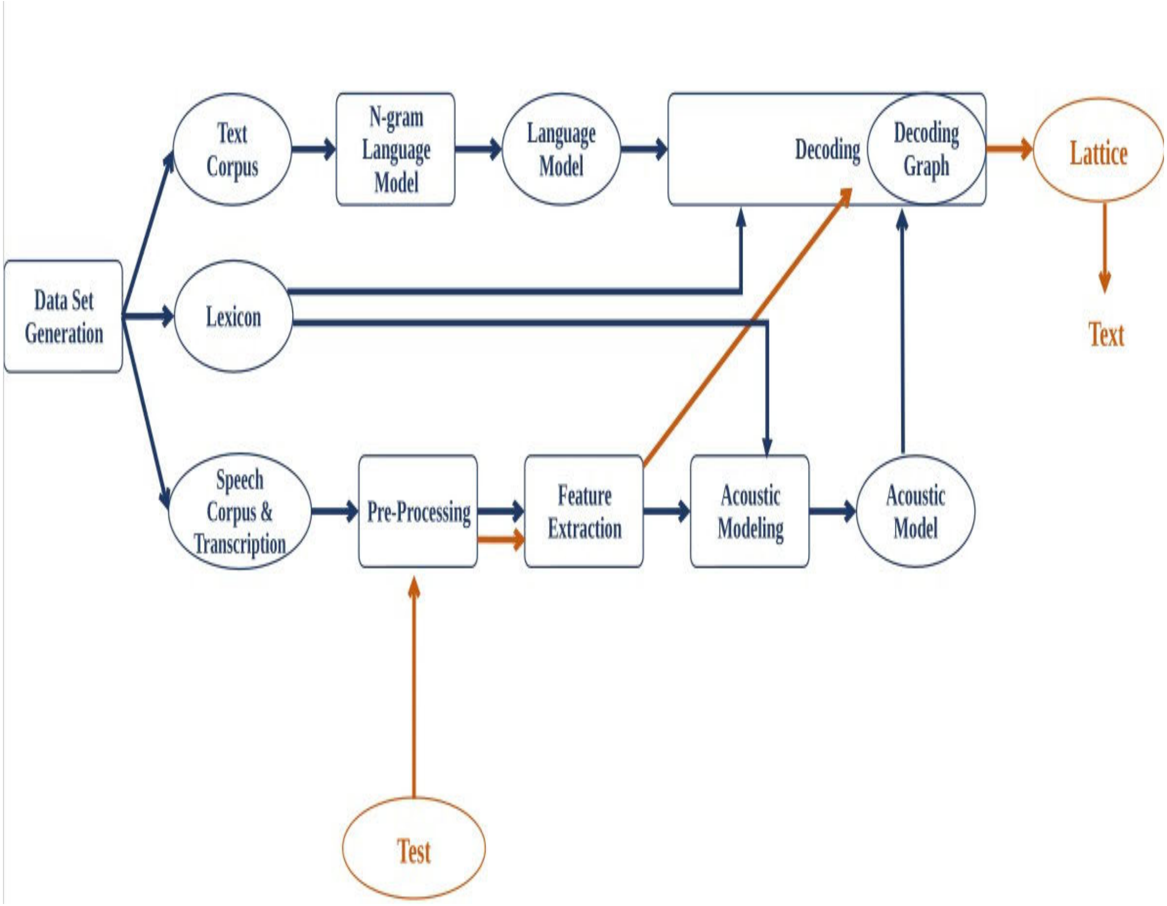


FIGURE 9. Overall methodology.

TABLE 1. Speech corpus technical specification.

CodeC:	WAV
Sample Rate:	16000 Hz
Bit Rate:	256 kbps
Channels:	Mono

auxiliary scripts that make it possible to create and test statistical language models for speech recognition and other applications [66].

D. TRAINING ACOUSTIC MODELS AND DECODING

We conducted two experiments to train the acoustic models from noise-free recordings and from noisy recordings, to decode the test data and to evaluate the ASR model. A variety of training strategies were used in these experiments to get more accurate findings and better-quality models. We trained the acoustic model with different models of GMM, SGMM and DNN.

1) EXPERIMENT 3: DECODING NOISE-FREE DATA

Experiment 3 was conducted to train acoustic models, and to subsequently decode noise-free data. The following are the steps of the experiment:

1) We divided the noise-free speech data into 5 folds and trained the models 5 times - each with an 80:20 Training: Testing ratio.
2) Pre-emphasis filtering was used as a pre-processing step.
3) MFCC Feature Extraction was used to extract 39 features. The waveform's amplitude was represented by 12 of the features, energy was represented by the 13th feature, the Δ-values among frames were represented by another 13 features, and the $\Delta - \Delta$ values were represented by the last 13 features.
4) Cepstral Mean & Variance Normalization was applied to the feature vectors. In doing so, the feature values were normalized by the mean, and divided by the variance.
5) The acoustic model was trained with different GMM, SGMM and DNN models.
6) After each training phase a Decoding Graph was generated.
7) The Acoustic Model was then used to decode the GMM, SGMM and DNN outputs.
8) After each training phase, the audio and text were aligned. This allowed the advanced training algorithms to use the values from each training phase to improve the parameters of the model.

TABLE 2. **GMM-HMM models parameters.**

Model Name	No. of Leaves	No. of Gaussians
Initial triphone model (tri1)	300	10000
$\Delta + \Delta\Delta$ (tri2a)	500	10000
LDA+MLLT (tri2B)	300	3000
LDA+MLLT+SAT (tri3B)	500	10000

For the GMM models we built a monophone model where information regarding each phone's prior and subsequent phones were neglected. The monophone model was then used as the foundation for triphone models. On top of it we built Delta and Delta Delta triphone models, which were trained using first and second order MFCC delta.

Then we built an LDA-MLLT triphone model, which was a highly refined model, where Linear Discriminant Analysis used feature vectors and created HMM states, but with a smaller feature space for all input, and the Maximum Likelihood Linear Transform used the LDA's reduced feature space to create a unique transformation for each speaker.

Next, we trained discriminative training algorithms. MMI and MPE was implemented on top of LDA+MLLT. SAT was implemented on top of LDA+MLLT to normalize noise and speakers, using a data transform for each speaker. Next, we trained a discriminative training algorithm on top of the LDA+MLLT+SAT.

The parameters, the number of Gaussians and the number of leaves, were specified at each call of the triphone training scripts. The number of Gaussians is the number of mixture models that the training should strive towards, while the number of leaves is the number of leaf nodes that should be targeted during the state tying process. There is no standard rule for determining the appropriate number of Gaussians or leaves. These values vary according to the type of the training data, and are determined through testing. Table 2 shows the parameters for each GMM-HMM training model.

For SGMM training, on top of the LDA+MLLT+SAT model, a Subspace Gaussian Mixture Model was trained, where the HMM states, or the decision tree leaves, have a same structure (i.e. the same number) of Gaussians. The number of Gaussians set for the HMM states was 700, the number of leaves was 500, and the number of substates was 1000.

The DNN model had two hidden layers. The p-norm function was used to activate the nodes and produce their output. The equation below shows the p-norm function [67].

$$y = \|x\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$$

Research in [67] showed that the p-norm function outperformed other activations (including various versions of Maxout, and tanh and ReLU) on a consistent basis. The p-norm units performed better, even with fewer parameters and layers. The p-norm input dimension was 1000 nodes, and the p-norm output dimension was 200 nodes, the initial learning rate was 0.02 and the final learning rate was 0.004. The input features are MFCC + LDA + MLLT + SAT. Fig. 10 illustrates the relations among these methods.

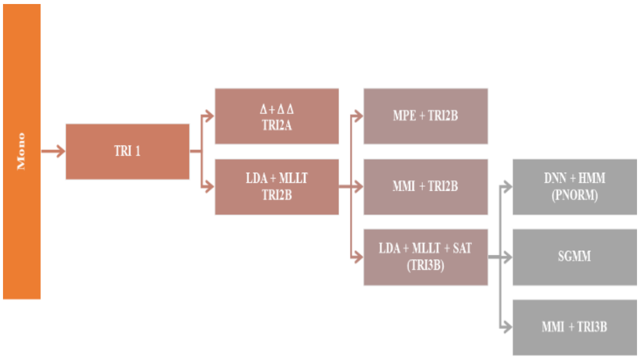


FIGURE 10. **Relations among models.**

2) EXPERIMENT 4: DECODING NOISY DATA

Experiment 4 was conducted to train with and decode noisy data. The steps for this experiment are the same as Experiment 3 steps, except that we use noisy data.

E. EXPERIMENT 5: USABILITY STUDY

This study goal was to investigate the usefulness of the proposed car assistant system, and to evaluate the performance of our Arabic speech recognition system as a part of the human-vehicle interaction framework. This usability study was conducted using the following steps:

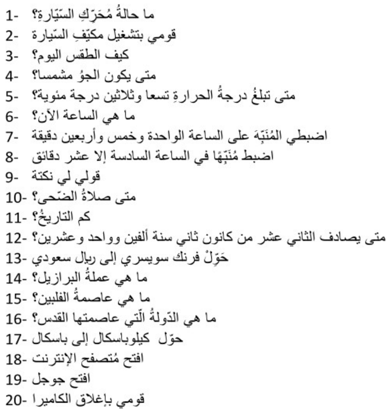
1) We chose 20 sentences from all topics in the system's lexicon. Fig. 11 shows the selected sentences.
2) We developed an evaluation sheet, where 5 indicated strong agreement and 1 indicated significant disagreement with the statement. Table 3 shows the evaluation sheet.
3) Ten car drivers were chosen to perform this experiment, ranging from novice to expert drivers. Participants in the study ranged in age from 20 to 75.
4) Participants were asked to read the 20 sentences twice, with and without traffic background noise.
5) These readings were then decoded in real time, where the participants could see the recognized words immediately. Realtime decoding meant that the participant did not have to wait until all the audio was acquired.
6) After reading the 20 sentences twice, each participant was asked to complete the evaluation sheet.

V. RESULTS AND DISCUSSION

A. DATA SET

1) EXPERIMENT 1: GENERATING THE LEXICON

Experiment 1 results are as follows. The prepared lexicon contained about 4000 words, covering a range of topics, including car parts, weather, currencies, dates, units, prayers, capitals, cities, and application services. Fig. 12 shows some of the words from the lexicon.

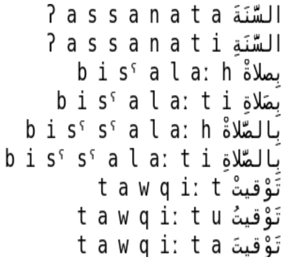


1- ما حالةُ مُحَرّكِ السّيّارةِ؟
2- قومي بتشغيل مكيّفِ السّيارة
3- كيف الطقس اليوم؟
4- متى يكون الجوُ مشمسا؟
5- متى تبلغُ درجةُ الحرارةِ تسعا وثلاثين درجة مئوية؟
6- ما هي الساعة الآن؟
7- اضبطي المُنَبّهَ على الساعة الواحدة وخمس وأربعين دقيقة
8- اضبط مُنَبّهَا في الساعة السادسة إلا عشر دقائق
9- قولي لي نكتة
10- متى صلاةُ الضّحى؟
11- كم التاريخُ؟
12- متى يصادف الثاني عشر من كانون ثاني سنة ألفين وواحد وعشرين؟
13- حَوّلْ فرنك سويسري إلى ريال سعودي
14- ما هي عملةُ البرازيل؟
15- ما هي عاصمةُ الفلبين؟
16- ما هي الدّولةُ الّتي عاصمتها القدس؟
17- حوّل كيلوباسكال إلى باسكال
18- افتح مُتصفح الإنترنت
19- افتح جوجل
20- قومي بإغلاق الكاميرا

FIGURE 11. Selected sentences for usability study.

TABLE 3. Evaluation sheet.

#	Item	Score
1	**Responsiveness**: ASR system is responsive	1 2 3 4 5
2	**Adequate**: The fields included in the system's lexicon are adequate	1 2 3 4 5
3	**Frequency of Use**: The system is frequently used	1 2 3 4 5
4	**Satisfaction**: Satisfied with the recognized words	1 2 3 4 5
5	**Importance**: The system is important	1 2 3 4 5
6	**Noise Tolerance**: The System can tolerate noise	1 2 3 4 5
7	**Driver Distraction**: The system will reduce the driver distraction	1 2 3 4 5
8	**Ease of Use**: The system is easy to use	1 2 3 4 5
9	**Purchase**: Users are willing to buy the system, if it was available for sale	1 2 3 4 5
10	**System Rating**: System overall rating	1 2 3 4 5



```
ʔ a s s a n a t a   السّنَةَ
ʔ a s s a n a t i   السّنَةِ
b i sˤ a l a: h   بِصلاةْ
b i sˤ a l a: t i   بِصَلاةِ
b i sˤ sˤ a l a: h   بِالصّلاةْ
b i sˤ sˤ a l a: t i   بِالصّلاةِ
t a w q i: t   تَوْقيتْ
t a w q i: t u   تَوْقيتُ
t a w q i: t a   تَوْقيتَ
```

FIGURE 12. Words from Lexicon.

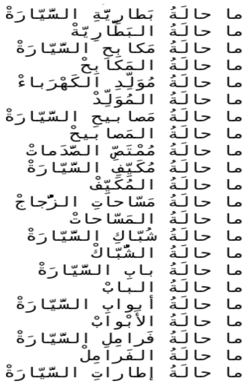
2) EXPERIMENT 2: GENERATING THE CORPUS

The text corpus was comprised of 6110 elements, including words, phrases, and sentences. According to our experiment, asking about the time, weather, locations, the status of car parts, prayer times, giving some orders like turn AC on or off, open car window, turn on music, raise volume and others, are primarily used in Human-Vehicle Interaction. Fig. 13 shows a small portion of the text corpus.

Experiment 2 results were as follows. The speech corpus had more than 60,000 recordings (almost 50 recording hours) along with their transcriptions, 30,000 recordings (almost 25 recording hours) without traffic noise, and 30,000 recordings with traffic noise.

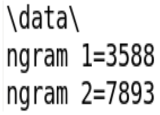
B. LANGUAGE MODEL

The output of the SRILM toolkit is a file that contains a language model in ARPA format, where various probabilities of an n-gram are listed. Fig. 14 shows the header of the ARPA file format, which shows the number of distinct n-gram types detected for each order n, up to the model's maximum order. Fig. 15 shows a sample list of 2-grams.



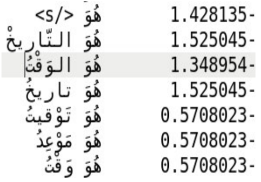
ما حالَةُ بَطارِيّةِ السّيّارَةْ
ما حالَةُ البَطّارِيّةْ
ما حالَةُ مَكابِحِ السّيّارَةْ
ما حالَةُ المَكابِحْ
ما حالَةُ مُوَلِّدِ الكَهْرَباءْ
ما حالَةُ المُوَلِّدْ
ما حالَةُ مَصابيحِ السّيّارَةْ
ما حالَةُ المَصابيحْ
ما حالَةُ مُمْتَصِّ الصّدَماتْ
ما حالَةُ مُكَيِّفِ السّيّارَةْ
ما حالَةُ المُكَيِّفْ
ما حالَةُ مَسّاحاتِ الزّجاجْ
ما حالَةُ المَسّاحاتْ
ما حالَةُ شُبّاكِ السّيّارَةْ
ما حالَةُ الشّبّاكْ
ما حالَةُ بابِ السّيّارَةْ
ما حالَةُ البابْ
ما حالَةُ أبوابِ السّيّارَةْ
ما حالَةُ الأبْوابْ
ما حالَةُ فَرامِلِ السّيّارَةْ
ما حالَةُ الفَرامِلْ
ما حالَةُ إطاراتِ السّيّارَةْ

FIGURE 13. Part of text corpus.



```
\data\
ngram 1=3588
ngram 2=7893
```

FIGURE 14. Header of ARPA Format LM.



```
<s/> هُوَ        1.428135-
التّاريخْ هُوَ     1.525045-
الوَقْتُ هُوَ      1.348954-
تاريخُ هُوَ       1.525045-
تَوْقيتُ هُوَ      0.5708023-
مَوْعِدُ هُوَ      0.5708023-
وَقتُ هُوَ       0.5708023-
```

FIGURE 15. List of 2-grams.

Each of the 2-grams is followed by the log (base-10) of the word's conditional probability given the previous 1-gram words. This says that the probability of the word (attareekh) coming after (Howa) is $10^{-1.525045} = 0.029850733$.

TABLE 4. Experiment 3 results.

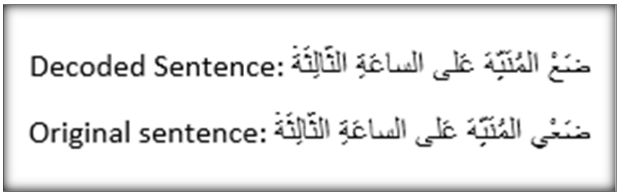
Model	WER (%)
Monophone	11.038
tri1	8.442
tri2a	8.522
tri2B	8.94
MMI+tri2B	10.252
MPE+tri2B	8.658
tri3B	5.87
MMI+tri3B	6.524
SGMM	5.80
DNN-HMM	5.168

C. TRAINING ACOUSTIC MODELS AND DECODING

1) EXPERIMENT 3: DECODING NOISE-FREE DATA

Table 4 shows the average WERs from Experiment 3.

The average WER for the monophone models was the worst among all the models. This was expected because context-independent phones were used in building this model. The first large improvement can be seen when switching from monophones to triphones as the unit used in training, where the WER decreased by 2.596%. Tri2B (LDA+MLLT) performed slightly worse. This was not expected, but was



Decoded Sentence: ضَبَعْ المُنَبِّهَ عَلى السّاعَةِ الثّالِثَة

Original sentence: ضَبَعِي المُنَبِّهَ عَلى السّاعَةِ الثّالِثَة

FIGURE 16. Example of incorrect decoded sentence.

TABLE 5. Experiment 4 results.

Model	WER (%)
Monophone	14.49
tri1	11.374
tri2a	11.62
tri2B	12.52
MMI+tri2B	11.928
MPE+tri2B	11.476
tri3B	8.18
MMI+tri3B	8.3
SGMM	7.38
DNN-HMM	6.684

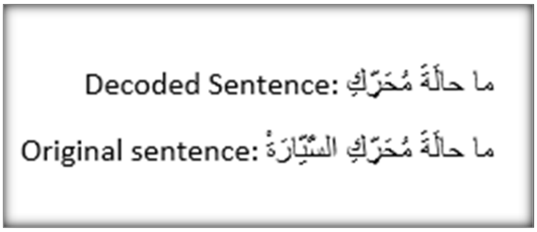
attributed to the small amount of training data, which did not allow the system to do adequate alignments. MPE+tri2B performed slightly better, while MMI+tri2B performed worse, which was also attributed to the small amount of data. The best result for the acoustic model (trained using GMM-HMM) was obtained for the tri3B model with a 5.87% WER. That's a 3.07% WER improvement from the previous model. SGMM also showed an improvement with a 5.80% WER, and the best of all models was the DNN-HMM model with a 5.168% WER. For DNN-HMM model most of the errors are coming from the masculine and feminine suffixes, word prefixes and some letters' diacritics. Fig. 16 shows an example of sentence decoded incorrectly, where masculine verb decoded instead of feminine one.

2) EXPERIMENT 4: DECODING NOISY DATA

Table 5 shows the average WERs from Experiment 4.

In Experiment 4, there was no pre-processing step for noise cancellation. Its purpose was to test how well the model would recognize noisy data.

The monophone model WER was the worst performing among all the models with a 14.49% WER. The first significant improvement was seen when switching from monophones to triphones with a 3.116% improvement in WER. The best triphone result was from the tri3B model (i.e. the acoustic model trained using GMM-HMM) with an 8.18% WER which was a 4.34% WER improvement from the previous triphone model. SGMM showed a further improvement with a 7.38% WER, and the best of all models was the DNN-HMM model with a 6.684% WER. For DNN-HMM model most of the errors are coming from the masculine and feminine suffixes, word prefixes, some letters' diacritics and missing some words in some cases when noise were high. Fig. 17 shows an example of sentence decoded incorrectly, where one word is missed in the decoded sentence.



Decoded Sentence: ما حالَةُ مُحَرِّكِ

Original sentence: ما حالَةُ مُحَرِّكِ السَّيّارَة

FIGURE 17. Example of incorrect decoded sentence.

TABLE 6. Experiment 5 results.

#	Item	Mean Score
1	**Responsiveness**: ASR system is responsive	4.7
2	**Adequate**: The fields included in the system's lexicon are adequate	4.1
3	**Frequency of Use**: The system is frequently used	4.4
4	**Satisfaction**: Satisfied with the recognized words	4.2
5	**Importance**: The system is important	4.9
6	**Noise Tolerance**: The System can tolerate noise	3.7
7	**Driver Distraction**: The system will reduce the driver distraction	4.5
8	**Ease of Use**: The system is easy to use	5
9	**Purchase**: Users are willing to buy the system, if it was available for sale	4.3
10	**System Rating**: System overall rating	4.1

D. EXPERIMENT 5: USABILITY STUDY

Table 6 shows the results from Experiment 5, in the form of the mean scores from the evaluation sheets filled out by ten participants after they read the sentences and saw the transcriptions.

In general, these results are encouraging. Except for ''Noise Tolerance'', all the criteria used to measure the system's usability scored mean ratings of more than 4.0 out of 5.0. The highest score was for ''Ease of Use'', where no prior experience was needed to use the system. The lowest score was for ''Noise Tolerance'', which indicates that the system did not perform very well within noisy inputs. This motivates us to concentrate our efforts on improving the system's performance when dealing with noisy data, via the use of noise filters.

VI. CONCLUSION AND FUTURE WORK

A. CONCLUSION

This research proposed to answer the following research questions:

Q1: How could we develop an automatic speech recognition system for an Arabic car assistant?

A: This study proposed a car assistant speech recognizer for Arabic language with the aim of reducing driver distraction to reduce car accident rates. To achieve our goal, MFCC was used to extract the acoustic features, and a bi-gram language model was analyzed using SRILM toolkit. Three different acoustic models GMM-HMM, SGMM-HMM and DNN-HMM were developed twice (with and without noise) and compared. For the GMM-HMM approach, various discriminative and adaptation techniques were trained at different stages.

The experimental results showed that the Deep Neural Network (DNN) outperformed the other models with a 5.168% WER, while SGMM-HMM had a 5.80% WER, and the best GMM-HMM model had a 5.87% WER, for noise-free data. For noisy data DNN-HMM had a 6.684% WER, SGMM-HMM had a 7.38% WER, and the best GMM-HMM had an 8.18% WER. It was also shown that MPE discriminative training performs better than MMI for this data. This system was judged by 10 participants to be important, responsive and able to reduce driver distraction.

Q2: How can we develop a lexicon, as well as text and speech corpora in Arabic specifically for the car assistant domain?

A2: This study built a domain-specific lexicon, to comprehensively support the human-vehicle interaction domain. Many linguistic variants were transcribed using IPA to produce comprehensive text and speech corpora, and this data was used successfully to build our system. This data will be made available to other researchers upon request.

B. FUTURE WORK

To improve our system performance, we must extend the topics included in the lexicon, and extend the speech corpus to hundreds, or even thousands, of hours. We must also use effective noise filters as a pre-processing step. Based on the literature and our experimental results, it is anticipated that building an end-to-end ASR system using a Time Delay Neural Network (TDNN) will achieve better recognition rates.

REFERENCES

[1] K. Young, M. Regan, and M. Hammer, "Driver distraction: A review of the literature," in *Distracted Driving*. 2007, pp. 379–405.

[2] D. S. Hurwitz, E. Miller, M. Jannat, L. N. Boyle, S. Brown, A. Abdel-Rahim, and H. Wang, "Improving teenage driver perceptions regarding the impact of distracted driving in the Pacific northwest," *J. Transp. Saf. Secur.*, vol. 8, no. 2, pp. 148–163, Apr. 2016.

[3] F. Weng, P. Angkititrakul, E. E. Shriberg, L. Heck, S. Peters, and J. H. L. Hansen, "Conversational in-vehicle dialog systems: The past, present, and future," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 49–60, Nov. 2016.

[4] U. Kamath, J. Liu, and J. Whitaker, "*Deep Learning for NLP and Speech Recognition*, vol. 84. Cham, Switzerland: Springer, 2019.

[5] R. E. Gruhn, W. Minker, and S. Nakamura, *Statistical Pronunciation Modeling for Non-Native Speech Processing*. Berlin, Germany: Springer, 2011.

[6] V. R. Gil, "Automatic speech recognition with Kaldi toolkit," B.S. thesis, Dept. Sci. Telecommun. Technol. Eng., Universitat Politècnica de Catalunya, Barcelona, Spain, 2016.

[7] S. Alyousefi, "Digital automatic speech recognition using Kaldi," Ph.D. thesis, Florida Inst. Technol., Melbourne, FL, USA, 2018.

[8] N. Thieberger, "What remains to be done—Exposing invisible collections in the other 7,000 languages and why it is a DH enterprise," *Digit. Scholarship Hum.*, vol. 32, no. 2, pp. 423–434, 2017.

[9] H. Ibrahim and A. Varol, "A study on automatic speech recognition systems," in *Proc. 8th Int. Symp. Digit. Forensics Secur. (ISDFS)*, Jun. 2020, pp. 1–5.

[10] J. Kaur, A. Singh, and V. Kadyan, "Automatic speech recognition system for tonal languages: State-of-the-art survey," *Arch. Comput. Methods Eng.*, vol. 28, pp. 1039–1068, Feb. 2020.

[11] A. Rafalovitch and R. Dale, "United Nations general assembly resolutions: A six-language parallel corpus," in *Proc. MT Summit*, vol. 12, 2009, pp. 292–299.

[12] M. A.-A. M. Abushariah, R. N. Ainon, R. Zainuddin, M. Elshafei, and O. O. Khalifa, "Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus," *Int. Arab J. Inf. Technol.*, vol. 9, no. 1, pp. 84–93, 2012.

[13] A. Abdelali, "Localization in modern standard Arabic," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 55, no. 1, pp. 23–28, 2004.

[14] B. H. A. Ahmed and A. S. Ghabayen, "Arabic automatic speech recognition enhancement," in *Proc. Palestinian Int. Conf. Inf. Commun. Technol. (PICICT)*, May 2017, pp. 98–102.

[15] H. Hyassat and R. A. Zitar, "Arabic speech recognition using SPHINX engine," *Int. J. Speech Technol.*, vol. 9, nos. 3–4, pp. 133–150, Dec. 2006.

[16] H. Satori, M. Harti, and N. Chenfour, "Arabic speech recognition system based on CMUSphinx," in *Proc. Int. Symp. Comput. Intell. Intell. Informat.*, Mar. 2007, pp. 31–35.

[17] A. Ali, Y. Zhang, P. Cardinal, N. Dahak, S. Vogel, and J. Glass, "A complete KALDI recipe for building Arabic speech recognition systems," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2014, pp. 525–529, doi: 10.1109/SLT.2014.7078629.

[18] D. Vergyri, A. Mandal, W. Wang, A. Stolcke, J. Zheng, M. Graciarena, D. Rybach, C. Gollan, R. Schlüter, K. Kirchhoff, A. Faria, and N. Morgan, "Development of the SRI/nightingale Arabic ASR system," in *Proc. 9th Annu. Conf. Int. Speech Commun. Assoc.*, Sep. 2008, pp. 1437–1440.

[19] M. M. Azmi, H. Tolba, S. Mahdy, and M. Fashal, "Syllable-based automatic Arabic speech recognition," in *Proc. WSEAS Int. Conf. Signal Process., Robot. Autom. (ISPRA)*, 2008, pp. 246–250.

[20] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[21] E. Koffi, "A tutorial on acoustic phonetic feature extraction for automatic speech recognition (ASR) and text-to-speech (TTS) applications in African languages," *Linguistic Portfolios*, vol. 9, no. 1, p. 11, 2020.

[22] S. J. Arora and R. P. Singh, "Automatic speech recognition: A review," *Int. J. Comput. Appl.*, vol. 60, no. 9, pp. 1–11, 2012.

[23] S. Karpagavalli and E. Chandra, "A review on automatic speech recognition architecture and approaches," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 9, no. 4, pp. 393–404, Apr. 2016.

[24] M. Alshayeji, M. Alshayeji, and S. Sultan, "Diacritics effect on Arabic speech recognition," *Arabian J. Sci. Eng.*, vol. 44, no. 1, pp. 9043–9056, 2019.

[25] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, "The subspace Gaussian mixture model—A structured model for speech recognition," *Comput. Speech Lang.*, vol. 25, no. 2, pp. 404–439, 2011.

[26] B. D. Sarma and S. M. Prasanna, "Acoustic–phonetic analysis for speech recognition: A review," *IETE Tech. Rev.*, vol. 35, no. 3, pp. 305–327, 2018.

[27] V. Bhardwaj, S. Bala, V. Kadyan, and V. Kukreja, "Development of robust automatic speech recognition system for Children's using kaldi toolkit," in *Proc. 2nd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2020, pp. 10–13.

[28] L. Li, Y. Zhao, D. Jiang, Y. Zhang, F. Wang, I. Gonzalez, E. Valentin, and H. Sahli, "Hybrid deep neural network-hidden Markov model (DNN-HMM) based speech emotion recognition," in *Proc. Hum. Assoc. Conf. Affect. Comput. Intell. Interact.*, Sep. 2013, pp. 312–317, doi: 10.1109/ACII.2013.58.

[29] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," 2014, *arXiv:1412.5567*.

[30] B. Liu, W. Zhang, X. Xu, and D. Chen, "Time delay recurrent neural network for speech recognition," *J. Phys., Conf. Ser.*, vol. 1229, no. 1, May 2019, Art. no. 012078.

[31] M. A. A. Aung and W. P. Pa, "Time delay neural network for Myanmar automatic speech recognition," in *Proc. IEEE Conf. Comput. Appl. (ICCA)*, Feb. 2020, pp. 1–4.

[32] Z. Hachkar, A. Farchi, B. Mounir, and J. E. Abbadi, "A comparison of DHMM and DTW for isolated digits recognition system of Arabic language," *Int. J. Comput. Sci. Eng.*, vol. 3, no. 3, pp. 1002–1008, 2011.

[33] A. R. Ali, "Multi-dialect Arabic speech recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–7.

[34] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, and J. Allen, "SPEECHDAT-CAR. A large speech database for automotive environments," in *Proc. LREC*, 2000, pp. 1–6.

[35] J. H. L. Hansen, P. Angkititrakul, J. Plucienkowski, S. Gallant, U. Yapanel, B. Pellom, W. Ward, and R. Cole, "'CU-move': Analysis & corpus development for interactive in-vehicle speech systems," in *Proc. 7th Eur. Conf. Speech Commun. Technol. (Eurospeech)*, Sep. 2001, pp. 1–4.

[36] H. Nacereddine, ‘‘Contribution to the automatic speech recognition of Arabic language and its applications,’’ Ph.D. thesis, Université Badji Mokhtar, Annaba, Algeria, 2014.

[37] Y. A. Ibrahim, J. C. Odiketa, and T. S. Ibiyemi, ‘‘Preprocessing technique in automatic speech recognition for human computer interaction: An overview,’’ *Ann. Comput. Sci. Ser.*, vol. 15, no. 1, pp. 186–191, 2017.

[38] M. F. Font, ‘‘Multi-microphone signal processing for automatic speech recognition in meeting rooms,’’ M.S. thesis, Augmented Multi-Party Interact. (AMI), Institut Dalle Molle d’Intelligence Artificielle Perceptive (IDIAP), Univ. Edimburgh, Edimburgh, U.K., 2005.

[39] D. Gupta, P. Bansal, and K. Choudhary, ‘‘The state of the art of feature extraction techniques in speech recognition,’’ in *Speech and Language Processing for Human-Machine Communications*. Switzerland: Springer, 2018, pp. 195–207.

[40] N. Desai, K. Dhameliya, and V. Desai, ‘‘Feature extraction and classification techniques for speech recognition: A review,’’ *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 12, pp. 367–371, 2013.

[41] K. Gupta and D. Gupta, ‘‘An analysis on LPC, RASTA and MFCC techniques in automatic speech recognition system,’’ in *Proc. 6th Int. Conf. Cloud Syst. Big Data Eng. (Confluence)*, Jan. 2016, Art. no. 493497.

[42] V. Tunalı, ‘‘A speaker dependent, large vocabulary, isolated word speech recognition system for Turkish,’’ M.S. thesis, Dept. Comput. Eng., Marmara Univ., İstanbul, Turkey, 2005.

[43] M. Ursin, ‘‘Triphone clustering in continuous speech recognition,’’ M.S. thesis, Dept. Comput. Sci., Helsinki Univ. Technol., Espoo, Finland, 2002.

[44] V. Keselj, D. Jurafsky, and J. H. Martin, *Speech and Language Processing*. London, U.K.: Pearson, 2009, p. 988.

[45] J. G. Kemeny and J. L. Snell, *Markov Chains*. vol. 6. New York, NY, USA: Springer-Verlag, 1976.

[46] L. R. Rabiner, ‘‘A tutorial on hidden Markov models and selected applications in speech recognition,’’ *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[47] E. Kullmann, ‘‘Speech to text for Swedish using KALDI,’’ School Eng. Sci. (SCI), Dept. Math., Optim. Syst. Theory, KTH, Sweden, Tech. Rep., 2016.

[48] N. Gabriel, ‘‘Automatic speech recognition in Somali,’’ Division Statist. Mach. Learn., Dept. Comput. Inf. Sci., Linköping Univ., Sweden, Tech. Rep., 2020.

[49] D. A. Reynolds, ‘‘Gaussian mixture models,’’ in *Encyclopedia of Biometrics*, vol. 741. 2009, pp. 659–663.

[50] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, ‘‘Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,’’ *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[51] D. Yu and L. Deng, *Automatic Speech Recognition*. Springer, 2016.

[52] M. Gales and S. Young, ‘‘The application of hidden Markov models in speech recognition,’’ Cambridge Univ., Cambridge, U.K., Tech. Rep., 2008.

[53] T. D. Gunnarsson, ‘‘Speech recognition for telephone conversations in Icelandic using Kaldi,’’ School Elect. Eng. Comput. Sci. (EECS), KTH, Sweden, Tech. Rep., 2019.

[54] P. C. Woodland and D. Povey, ‘‘Large scale discriminative training of hidden Markov models for speech recognition,’’ *Comput. Speech Lang.*, vol. 16, no. 1, pp. 25–47, Jan. 2002.

[55] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, ‘‘Boosted MMI for model and feature-space discriminative training,’’ in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 4057–4060.

[56] K. Shinoda, ‘‘Speaker adaptation techniques for automatic speech recognition,’’ in *Proc. APSIPA ASC*, 2011, pp. 1–9.

[57] J. Ganitkevitch, ‘‘Speaker adaptation using maximum likelihood linear regression,’’ Rheinish-Westflesche Technische Hochschule Aachen, Aachen, Germany, Tech. Rep., 2005. [Online]. Available: https://www-i6.informatik.rwthaachen.de/web/Teaching/Seminars/SS05Ganitkevitch Ausarbeitung.pdf

[58] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, ‘‘The Kaldi speech recognition toolkit,’’ in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. Conf.* Piscataway, NJ, USA: IEEE Signal Processing Society, Dec. 2011, pp. 1–4.

[59] M. Mohri, F. Pereira, and M. Riley, ‘‘Weighted finite-state transducers in speech recognition,’’ *Comput. Speech Lang.*, vol. 16, no. 1, pp. 69–88, Jan. 2002.

[60] M. Mohri, F. Pereira, and M. Riley, ‘‘Speech recognition with weighted finite-state transducers,’’ in *Springer Handbook of Speech Processing*. Switzerland: Springer, 2008, pp. 559–584.

[61] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiat, S. Kombrink, P. Motlicek, Y. Qian, K. Riedhammer, K. Vesely, and N. T. Vu, ‘‘Generating exact lattices in the WFST framework,’’ in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 4213–4216.

[62] X. Lingyun and D. Limin, ‘‘Efficient Viterbi beam search algorithm using dynamic pruning,’’ in *Proc. 7th Int. Conf. Signal Process. (ICSP)*, vol. 1, 2004, pp. 699–702.

[63] A. Brown, ‘‘International phonetic alphabet,’’ in *The Encyclopedia of Applied Linguistics*. 2012.

[64] C. F. Meyer, *English Corpus Linguistics: An Introduction* (Studies in English Language). ERIC, 2002.

[65] V. Z. Kepuska and P. Rojanasthien, ‘‘Speech corpus generation from DVDs of movies and TV series,’’ *J. Int. Technol. Inf. Manag.*, vol. 20, no. 1, p. 4, 2011.

[66] A. Stolcke, ‘‘SRILM—An extensible language modeling toolkit,’’ in *Proc. 7th Int. Conf. Spoken Lang. Process. (ICSLP)*, USA, Sep. 2002.

[67] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, ‘‘Improving deep neural network acoustic models using generalized maxout networks,’’ in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 215–219.

GHADEER A. JARADAT received the M.S. degree in computer engineering from Yarmouk University, in 2022. She is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, Khalifa University.

Her research interests include assistive technology and machine learning applications.

MOHAMMAD A. ALZUBAIDI (Member, IEEE) received the Ph.D. degree in computer science and engineering from the Ira Fulton School of Engineering, Arizona State University, in 2012.

He is currently an Associate Professor of computer engineering at Yarmouk University, Jordan. His research interests include medical imaging perception and understanding, computer vision, pattern recognition, assistive technology, and machine learning.

MWAFFAQ OTOOM (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Virginia Tech, in 2012.

He is a Full Professor of computer engineering at Yarmouk University, Jordan. His research interests include novel modeling techniques for embedded systems, assistive technology, and machine learning.