Comparative Study of Amazigh Speech Recognition Systems Based on Different Toolkits and Approaches

Safâa EL OUAHABI[1], *Sara* EL OUAHABI[2] *and Mohamed* ATOUNTI[1]

[1] Laboratory of Applied Mathematics and Information Systems, Multidisciplinary Faculty of Nador, Mohamed First University Oujda, Morocco
[2] SOVIA team, LSA, National School of Applied Sciences Hoceima, Abdelmalek Essaadi University of Tangier, 90000, Tangier, Morocco

Abstract. The objective of this study is to evaluate and contrast the performance of different ASR approaches applied to the Amazigh language. Markovian modelling techniques, including Hidden Markov Models with Gaussian mixture distribution, Convolutional Neural Network, size of vocabulary, and lastly, the choice of decoder, whether Sphinx or HTK, by conducting a comprehensive analysis and comparison of these factors, this paper aims to provide valuable insights into the development of effective ASR systems for the Amazigh language. The findings will contribute to advancing the field of Amazigh ASR and aid in the selection of appropriate techniques and tools for future research and development efforts.

1 Introduction

The use of speech as a mode of communication between humans and machines has gained significant importance in recent years. Speech provides a natural and intuitive means of interaction, enabling users to convey their intentions, commands, and queries in a seamless and efficient manner. This mode of communication has found extensive applications in various domains, including virtual assistants, customer service, voice-controlled devices, and automotive interfaces.

One of the key advantages of speech-based communication is its accessibility. It allows individuals with diverse abilities and literacy levels to interact with machines effortlessly. Speech interfaces eliminate the need for manual input, such as typing or navigating complex menu structures, making technology more inclusive and user-friendly.

Furthermore, speech offers a faster and more efficient way of conveying information compared to traditional input methods. It enables hands-free operation, freeing users from the constraints of physical devices and allowing them to perform tasks while on the move. This convenience is particularly valuable in scenarios where manual interaction is challenging or impractical, such as driving or multitasking.

Advancements in Automatic Speech Recognition (ASR) technology have played a pivotal role in enabling robust and accurate speech-based communication. ASR systems convert spoken language into text, facilitating seamless integration with machine learning algorithms and natural language processing techniques. These advancements have made substantial improvements in the accuracy and dependability of speech recognition, ultimately enhancing the overall user experience. However, challenges still exist in speech recognition, including handling variations in accents, background noise, and speech disorders. Ongoing research and development efforts aim to address these challenges and further enhance the performance of speech recognition systems.

Automatic speech processing covers a wide range of activities, often complementary, which can be classified into five main themes: (speech coding and compression, speech synthesis, speech recognition, speaker recognition, and verification, identification of the language in which a speaker is speaking). In all these areas, significant progress has been made in recent years and many industrial applications exist. We are interested in the problem of ASR for the Amazigh language. This problem is very difficult and complex, especially due to the characteristics of the speech signal of this language and its diversity.

In this study, our focus is to investigate the effectiveness of various tools for Amazigh ASR systems based on HMMs and CNNs. We will explore the capabilities of different toolkits, namely HTK and CMU Sphinx, in handling large vocabulary ASR systems for the Amazigh language. To begin, we provide a comprehensive review of the existing literature on Amazigh ASRs in Section 2. This literature review will cover the current state of research and advancements in Amazigh ASR technology. Section 3 describes the specific techniques and methodologies employed in building ASR systems for Amazigh. We will explore the utilization of HMMs and CNNs, highlighting their roles in acoustic modelling and feature extraction for Amazigh speech. Throughout the paper, we will evaluate the performance and capabilities of different toolkits, such as HTK and CMU Sphinx, in effectively handling the complexities and requirements of large vocabulary ASR systems for Amazigh. This analysis will provide insights into the strengths and limitations of each toolkit and their suitability for Amazigh ASR tasks. By the end of this study, we aim to contribute to the existing body of knowledge in the field of Amazigh ASR and provide recommendations for selecting appropriate toolkits and techniques for building robust and accurate ASR systems for the Amazigh language.

2 Literature review

In this section, we provide an overview of the existing literature on Automatic Speech Recognition systems developed for the Amazigh language. The following works represent some of the reported studies in this field:

[1] The objective of this study was to construct a dedicated ASR (Automatic Speech Recognition) system tailored for the Amazigh language. The system utilized an HMM (Hidden Markov Model) with Gaussian mixtures to establish the required acoustic models for accurate recognition. The researchers utilized the CMU Sphinx-4 tools for the development and evaluation of the system. During the experimentation phase, the system was trained and tested using various configurations. Different numbers of Gaussian mixtures, ranging from 1 to 256, were explored in conjunction with HMMs featuring 3 and 5 states. The performance of the system was evaluated based on recognition rates achieved.

[2] The authors have created an Amazigh numeral synthesis system based on isolated digits from 0 to 9. They modelled each phoneme as a 3-state hidden Markov model. The choice of this model is justified by its robustness for speech recognition applications. Additionally, their system, as mentioned by the author, serves as the first step of a security system.

[3] The authors have developed a speaker-independent speech recognition system. They based their research on the combination of several HMM and GMM using the CMU Sphinx

decoder. The system is tested on a dataset of Amazigh alphabets developed by the authors. The best results were obtained using a 5-state HMM with a 128-component GMM.

[4] The authors based their study on comparing HMM models with the dynamic programming method. The corpus used for the transcription is the Berber alphabet recognized by IRCAM. The results showed that the HMM method is more effective for speech recognition thanks to its stochastic modeling nature.

[5] The authors have compared 2 approaches HMM and CNN using the same dataset of 9240 audio, and they have obtained a recognition of 93,9%.

[6] The authors developed a recognition system using HTK. Their studies showed that the best score of 91.31% is obtained using a 32 Gaussian mixture distribution with a 3-state HMM. They calculated the recognition rate for each Amazigh digit and letter and subsequently combined all the alphabets to obtain a satisfactory recognition rate of 91.31%.

[7] The authors developed the first and only corpus for the Amazigh-Tarifit language, consisting of 187 words pronounced by 50 speakers, with 25% representing each gender. They developed their system using CMU Sphinx, which is based on HMM. The error rate is 8.20% through the combination of GMM and tied-state modeling.

[8] The authors tested the effect of triphone modeling and decision trees on a corpus of 187 Tarifit words pronounced by 50 speakers. The results achieved a recognition rate of 92.2%, which is the only score obtained in the literature for a corpus of 187 words. They used the CMU Sphinx decoder, which is based on HMM modeling.

These studies represent a subset of the available works in the literature that have focused on developing ASR systems for the Amazigh language. Each work contributes to the understanding and advancement of Amazigh ASR technology, addressing specific challenges and proposing novel methodologies.

It is important to note that this overview is not exhaustive, and there may be additional works and approaches in the field. The reported studies, however, provide valuable insights into the progress made in developing ASR systems tailored to the unique characteristics of the Amazigh language.

3 Toolkits and approach used for ASR

There are various toolkits and approaches used in speech recognition, including both traditional methods and modern deep-learning techniques. ASRs often utilize techniques such as data augmentation, beam search decoding, language models, and acoustic modeling techniques like adaptation or transfer learning to further improve performance. The specific choice of toolkit and approach depends on the task requirements, available data, and desired performance characteristics. In this section, we present the commonly used toolkits and approaches for Amazigh Language:

3.1 Hidden Markov models (HMMs) Speech recognition

consists of automatically transcribing spoken content in order to obtain the corresponding sequence of words. The first systems were able to transcribe only isolated words with a reduced vocabulary. During the last quarter of the 20th century, systems began to be able to transcribe continuous speech thanks to particular acoustic modeling based on HMM and stochastic modeling of language.

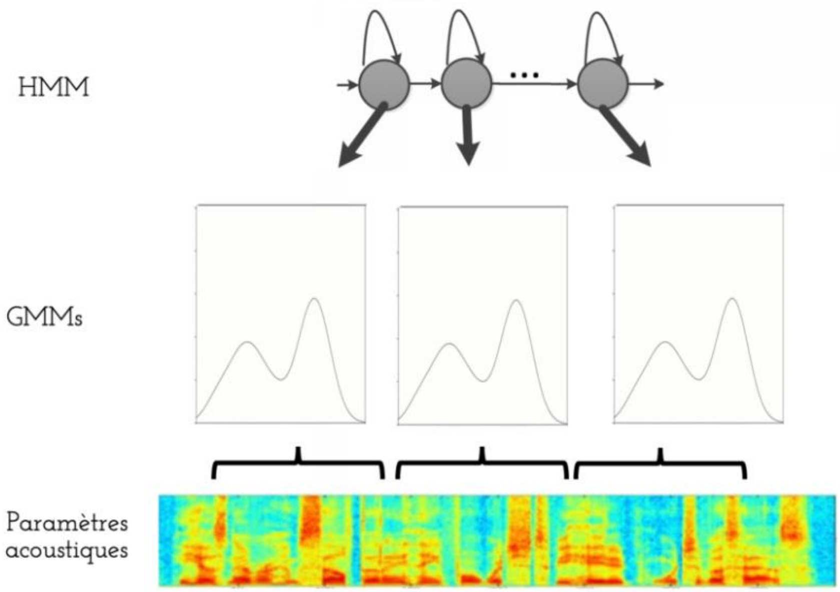
Two models are needed for this type of approach.

The acoustic model consists to recognize the sequences of phonemes (which form the words) present in a pronunciation dictionary. The training takes place over several hours of manually transcribed audio recordings.

The language model defines (on textual data) the probabilities of a (very large) set of possible word sequences.

A decoder (primarily, a graph search algorithm) integrates acoustic and linguistic knowledge to automatically transcribe the input recording. It should be noted that a feature extraction step is generally performed a priori in order to obtain a spectrum representation of these recordings' inputs.

The acoustic model represents the most important component of this architecture. It consists of a set of HMMs (generally) modelling phonemes whose emission probabilities are represented by mixtures of Gaussians (see Fig. 1. An HMM, therefore, estimates the probability of observing an acoustic form knowing a given phoneme.



*

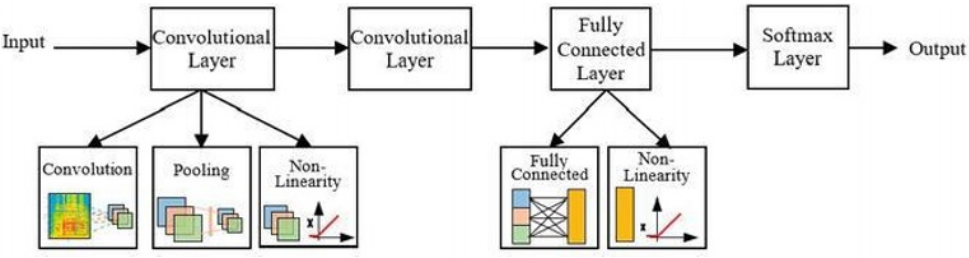
Fig. 1. HMM-GMM acoustic modelling.

3.2 Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) have demonstrated successful usage across numerous applications. Handwriting recognition, in particular, was among the initial applications where CNNs were effectively employed for image analysis [10]. Besides delivering excellent outcomes in tasks involving object detection and image classification [10, 11, 12], they also do well when applied to facial recognition [13, 14], video analysis [15, 16], or even text recognition [17, 18]. The principle of CNN is based on four key ideas that exploit the properties of natural signals [19]:

- Local connections, Shared weights.
- The pooling Layer.
- The Conventional Layer
- Fully connected Layer and SoftMax layer.

[*] https://www.aquiladata.fr/insights/commande-vocale-reconnaissance-automatique-de-la-parole/



†

Fig. 2. Block diagram of the convolutional neural network [20].

The architecture of a typical CNN is structured in a series of steps as mentioned in Fig.2. Here's a high-level overview of how CNNs can be used for speech recognition:

- Input Representation: The speech signal is typically pre-processed to obtain a time-frequency representation, such as a spectrogram or mel-frequency spectrogram. This representation divides the signal into frames and computes the magnitude of the Fourier transform or Mel-frequency cepstral coefficients (MFCCs) for each frame.
- Convolutional Layers: The CNN architecture consists of one or more convolutional layers. In each convolutional layer, a collection of adaptable filters is applied to the input representation. The filters are designed to capture local patterns in the spectrogram. Convolutional operations involve sliding the filters over the input representation and computing element-wise multiplications and summations. This process generates a set of feature maps that represent different learned features at various temporal and spectral resolutions.
- Non-linear Activation: After each convolutional operation, a non-linear activation function, such as ReLU (Rectified Linear Unit), is typically applied element-wise to introduce non-linearity into the network. This helps in capturing complex and non-linear relationships in speech data.
- Pooling Layers: Pooling layers are frequently utilized to decrease the dimensionality of the feature maps and introduce some degree of translation invariance. Max pooling is a commonly used pooling technique where the maximum value within a pooling window is retained, while other values are discarded. This downsamples the feature maps and makes the network more robust to small variations in the input.
- Fully Connected Layers: The convolutional and pooling layers' output is flattened and subsequently passed into one or more fully connected layers. These layers perform high-level feature extraction and map the learned features to the desired output classes or labels, such as phonemes or words. In most cases, the final layer utilizes a SoftMax activation function to generate a probability distribution across the output classes.
- Training: CNNs are trained using labelled speech data through a process called backpropagation, where the network's parameters are updated to minimize a chosen loss function. The most common loss function for speech recognition is the cross-entropy loss, which measures the dissimilarity between predicted probabilities and the true labels.

† https://www.intechopen.com/chapters/63017

- Decoding: During inference, the trained CNN can be used to recognize speech by applying a decoding algorithm, such as beam search or dynamic programming, to convert the network's output probabilities into a sequence of words or phonemes.

Convolutional Neural Networks (CNNs) have been successfully applied to various speech recognition tasks, particularly in the area of acoustic modelling. CNNs are effective in capturing local patterns and dependencies in spectral or spectrogram-like representations of speech signals.

3.3 Hidden Markov Model Toolkit (HTK)

HTK (Hidden Markov Model Toolkit) is a popular set of tools widely used for speech recognition and related tasks. It provides a comprehensive suite of tools for building and training HMM and performing various tasks in ASR research and development. Here are some key tools and functionalities offered by HTK as shown in Fig. 3:

- HCopy: HCopy is a tool used for speech feature extraction. It can convert speech waveforms into various feature representations, such as Mel-frequency cepstral coefficients (MFCCs), linear predictive coding (LPC) coefficients, or filter bank energies. HCopy supports various file formats and allows customization of feature extraction parameters.
- HCompV: HCompV is used for estimating the parameters of a Gaussian Mixture Model (GMM) from a set of speech feature vectors. It performs a preliminary clustering of the feature vectors and initializes the GMM's means and variances.
- HInit: HInit is used for initializing HMMs. It initializes the parameters of HMMs, such as the state transition probabilities and output probabilities, based on the given data. Initialization methods include discrete, uniform, and tied-state models.
- HRest: HRest is used for re-estimating the parameters of HMMs. It refines the HMM parameters based on the given training data. HRest can be used for both continuous density and discrete density HMMs.
- HMMIRest: HMMIRest is an extended version of HRest that supports semi-continuous and tied-mixture models. It allows the modelling of acoustic sub-phonetic units, which can help improve the modelling of speech units.
- HERest: HERest is used for training HMMs. It performs a sequence of re-estimations using the Baum-Welch algorithm, which is an iterative algorithm for maximum likelihood estimation. HERest is capable of handling multiple training data sets and supports various training configurations.

- HVite: HVite is used for decoding speech with HMMs. It performs recognition on a given set of feature vectors and produces the most likely sequence of HMM states. HVite supports various decoding modes, including forced alignment, Viterbi decoding, and lattice generation.
- HSGen: HSGen is a tool for generating synthetic speech using HMMs. It can generate speech waveforms based on a given HMM and text transcription.

These are just a few of the many tools provided by the HTK toolkit. HTK also offers utilities for training language models, performing various statistical analyses, and handling large speech corpora efficiently. It is a flexible and powerful toolkit that has been widely used in both research and industrial applications in the field of speech recognition.

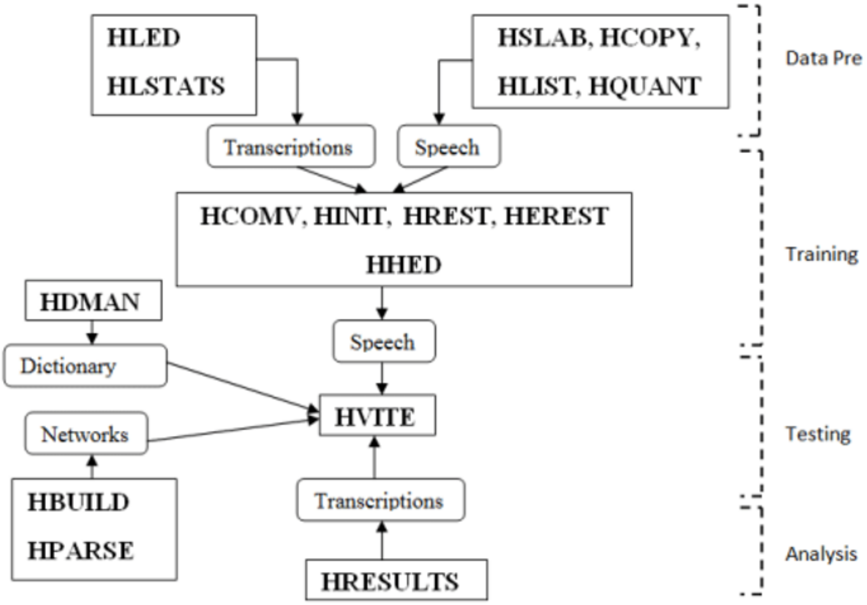


Fig. 3: Architecture of the development of an ASR using HTK [20].

3.4 CMU Sphinx Toolkit.

CMU Sphinx, also known as the Sphinx toolkit, is a collection of open-source speech recognition systems developed by Carnegie Mellon University (CMU). It provides a range of tools and libraries for building speech recognition applications. Here's an overview of CMU Sphinx and its components:

- Sphinx-4: Sphinx-4 is a flexible and modular Java-based speech recognition system. It provides a high-level API that allows developers to integrate speech recognition functionality into their Java applications. Sphinx-4 supports both offline and real-time speech recognition and includes various acoustic and language models.

- PocketSphinx: PocketSphinx is a lightweight and efficient speech recognition library written in C. It is designed for embedded and mobile platforms with limited computational resources. PocketSphinx offers real-time and continuous speech recognition capabilities and can be integrated into applications using C/C++ or Python.

- SphinxTrain: SphinxTrain is a set of tools for training acoustic and language models in the Sphinx framework. It includes utilities for preparing and labelling training data, creating pronunciation dictionaries, estimating acoustic model parameters, and building language models using n-grams or other statistical language modelling techniques.

- SphinxBase: SphinxBase is a common set of libraries and utilities shared by various components of CMU Sphinx. It provides functionalities for audio I/O, feature extraction, and common algorithms used in speech recognition, such as HMM decoding and language model handling.

- CMU Dictionary and Language Models: CMU Sphinx provides freely available language models and pronunciation dictionaries for use in speech recognition. These resources encompass various languages and domains, serving as a foundation for developing customized language models tailored to specific tasks.

- Sphinx Knowledge Base Tools (SphinxKB): SphinxKB is a set of tools and utilities for managing and manipulating knowledge bases for domain-specific speech recognition. It facilitates the creation and adaptation of language models and pronunciation dictionaries for specialized applications.

CMU Sphinx offers a range of capabilities, from offline large vocabulary continuous speech recognition (LVCSR) to embedded and mobile speech recognition. It has been widely used in both research and commercial applications. The toolkit provides a good balance between accuracy and computational efficiency, making it suitable for various speech recognition tasks and scenarios.

4 Comparative studies of Amazigh ASR

By examining and comparing the results of various experiments conducted on different Amazigh speech recognition systems cited in the literature reviews section, we conclude the followings comparison of results based on toolkits' difference: (HTK vs CMU Sphinx Tools) (see Table 1)

Table 1: Word recognition rates comparison using different Toolkits (Spoken digits and letters).

	[3]	**[2]**	**[6]**
CMU Sphinx	89.07	90	-
HTK	-	-	91.31

- In the context of Amazigh language recognition, the HTK Recognizer performs better than the CMU Sphinx Recognizer due to its superior modeling and precision learning.
- As detailed in [6], the HMM-GMM approach used for Amazigh language is robust, with a speaker-independent approach achieving a score of up to 91.31% using HMMs consisting of 3 states with a combination of 32 Gaussian Mixtures.
- With a small vocabulary consisting of only 10 digits and 33 letters, both word and phoneme models demonstrate high recognition accuracy. Representing a phone with a single HMM state is sufficient for recognizing a small number of words. The typical representation for a phone consists of three HMM states: a beginning state, a middle state, and an end state. Although it is technically feasible to use five states for a single phone, it has been demonstrated that this does not enhance accuracy as shown in Fig. 4.

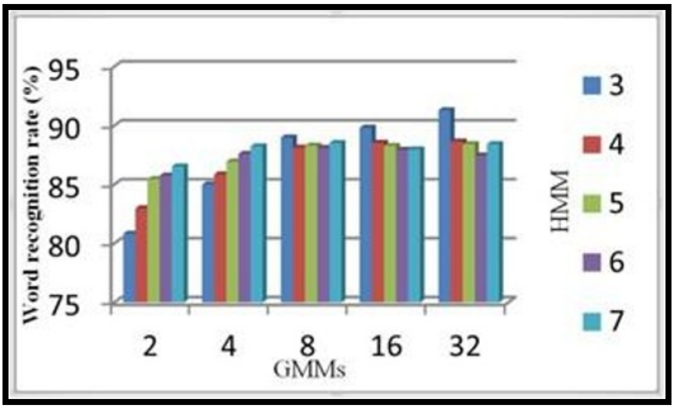


Fig. 4. Comparison of recognition rate (%) for different GMMs and HMMs using HTK Toolkit.

Table 2. Comparison of results based on approaches used: (HMM- GMM vs CNN).

	[5]	**[1]**	**[3]**	**[6]**	**[7]-[8]**
CNN	92			-	-
HMM-GMM	-	89.07	90	91.31	
HMM-GMM + Tied State	-	-		-	91.8
HMM-GMM + Tied State + Tree decision	-	-		-	92.2

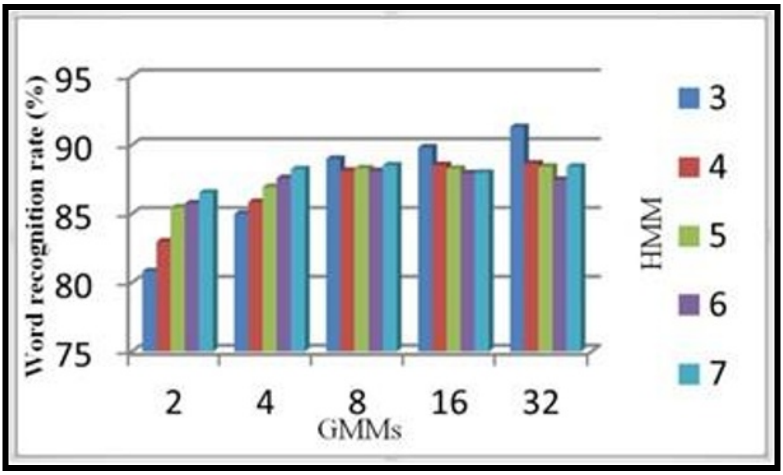


Fig. 5. Comparison of recognition rate (%) for different GMMs and HMMs using HTK Toolkit.

- [7] By incorporating additional samples of Amazigh words, the vocabulary and size of the databases have been expanded. (187 isolated words spoken by 50 speakers' width 5 repetitions). They have used the HMM-GMM approach to develop the system and added a parameter of tied state to improve the word recognition rate. Initially, the Word Error Rate (WER) for the Amazigh ASR system was relatively high, standing at 30 percent. This outcome was achieved utilizing the default values of the key parameters specified in the Sphinx configuration. To improve the system's performance, extensive testing and parameter adjustments were conducted. Specifically, modifications were made to the GMM and the tied states parameters. These modifications were implemented with the goal of improving the acoustic modeling and optimizing the system's capacity to accurately identify spoken words. As a result of these efforts, a significant reduction in the WER was achieved. The WER was successfully lowered to approximately 8, representing a substantial improvement. This reduction in error rate corresponds to a word recognition accuracy of 91.8 percent, indicating the system's enhanced capability to accurately transcribe Amazigh speech.

- Hidden Markov Models are used to pattern the acoustic states. Each state represents a phoneme or sub-phoneme unit, and the number of states can be quite large. With state-sharing using decision trees, the number of states can be significantly reduced by grouping similar states together. This reduces the overall complexity of the model and leads to more efficient computations.

- Gaussian Mixture Models (GMMs): GMMs are used to create the acoustic characteristics within each HMM state. Each state in the HMM is associated with a GMM that captures the distribution of acoustic feature vectors within that state. GMMs are used to estimate the probabilities of observing the acoustic features given the HMM state.

- State-Tying: State-tying is a technique used to group similar HMM states together, reducing the complexity of the model and improving efficiency. Tied states share the same GMM parameters, such as means and variances. State-tying is typically done based on phonetic similarity or acoustic similarity to group together states that exhibit similar acoustic characteristics.

- Decision trees are combined with tied states to further enhance the modeling process. A specific attribute is tested at each internal node, with the outcome represented by branches. Class labels or predicted values are assigned to leaf nodes.The decision tree helps guide the recognition process by determining the path through the tied states based on the observed features.

- Decision trees allow for the creation of powerful and expressive models by learning decision rules based on the input features. By sharing states, the decision tree can generalize well across different instances of similar states. This means that the model can capture and represent the common patterns and variations in speech, leading to improved recognition accuracy.

- The decision tree splits are learned during the training process by optimizing the acoustic likelihoods or error criteria. The decision tree guides the flow of the recognition process by selecting the most likely path through the tied states based on the observed acoustic features.

The findings presented in Table 2 and Fig. 5 allow us to draw meaningful conclusions regarding the accuracy of both the Hidden Markov Model-Gaussian Mixture Model (HMM-GMM) and Convolutional Neural Network (CNN) approaches in recognizing small vocabularies, specifically digits, and letters. For small vocabulary recognition tasks such as identifying 10 digits and 33 letters, the HMM-GMM approach demonstrates high accuracy. In this scenario, employing a single-state HMM to represent a phone is sufficient. The phone representation using HMM-GMM typically consists of three HMM states: the beginning, middle, and end states, combined with Gaussian Mixture Models. This configuration proves effective for accurately recognizing the given set of digits and letters.

On the other hand, the CNN approach has been investigated by the authors in a separate study [5] to enhance ASR for the Amazigh language. Notably, this study achieved an impressive accuracy rate of 93.9% using the CNN approach. The CNN architecture employed in the study effectively captured the acoustic features and temporal dynamics of the Amazigh language, resulting in highly accurate recognition outcomes.

These results indicate that both the HMM-GMM and CNN approaches can achieve high accuracy in recognizing small vocabularies, such as digits and letters. The choice between these approaches depends on the specific requirements of the recognition task and the resources available. The HMM-GMM approach with a single-state HMM may be suitable for simple vocabularies, while the CNN approach can provide enhanced performance for more complex languages like Amazigh. It is worth noting that the reported 93.9% accuracy rate in the study using the CNN approach for Amazigh ASR demonstrates the efficacy of the method in capturing the unique characteristics of the language and achieving remarkable recognition results. Overall, these findings shed light on the strengths and capabilities of both the HMM-GMM and CNN approaches in the context of small vocabulary recognition and highlight the remarkable accuracy achieved using CNN for Amazigh language ASR.

5 Conclusion

This paper provides a comprehensive analysis and comparison of various Automatic Speech Recognition (ASR) systems developed for the Amazigh language. The comparison focuses on two key approaches, namely HMM-GMM and CNN, taking into consideration factors such as vocabulary size and the type of decoder utilized (Sphinx and HTK).

For small vocabulary tasks, specifically recognizing digits and letters, both the HMM-GMM and CNN approaches demonstrate high recognition accuracy. These accuracies are achieved using either the CMU Sphinx or HTK tools. The experiments reveal that both approaches, in conjunction with these tools, effectively capture the acoustic and phonetic characteristics of the Amazigh language, resulting in accurate recognition outcomes.

However, when it comes to large vocabulary tasks, a notable improvement in recognition rates is observed. The use of a tree decision tree based on the HMM-GMM approach, incorporating tied states parameters, leads to a recognition rate of 92.2%. This indicates the significance of optimizing the system's architecture and parameters for large vocabulary tasks, resulting in enhanced recognition accuracy.

References

1. M. Telmem, Y. Ghanou, Estimation of the Optimal HMM Parameters for Amazigh Speech Recognition System Using CMU-Sphinx, proceedings of the first international conference on intelligent computing in data sciences, icds2017.
2. El Ghazi, C. DAOUI, N. IDRISSI, Automatic Speech Recognition for Tamazight Enchained Digit, World Journal Control Science and Engineering 2 (2014), no. 1, 1–5.
3. H. Satori, F. El Haoussi, Investigation Amazigh speech recognition using CMU tools, Int J Speech Technol 17, 235 (2014). https://doi.org/10.1007/s10772-014-9223-y.
4. Abenaou, F. Ataa Allah, B. Nsiri, Vers un systme de reconnaissance automatique de la parole amazighe bas´e sur les transformations orthogonales param´etrables. Asinag., 9 (2014), 133–145.
5. Telmem, Meryam, Ghanou, Youssef, A Comparative Study of HMMs and CNN Acoustic Model in Amazigh Recognition System, (2020) 10.1007/978-981-15-0947-6 50. https://doi.org/10.1016/j.procs.2018.01.102. 2018.
6. SAfâa El Ouahabi, Mohamed Atounti, Mohamed Bellouki. Optimal parameters selected for automatic recognition of spoken Amazigh digits and letters using Hidden Markov Model Toolkit, International Journal of Speech Technology (2020) 10.1007/s10772-020-09762-3.
7. SAfâa El Ouahabi, Mohamed Atounti, Mohamed Bellouki, Toward an automatic speech recognition system for amazigh-tarifit language. International Journal of Speech Technology, 22 (2019). 1–12. 10.1007/s10772-019-09617-6.
8. SAfâa El Ouahabi, Mohamed Atounti, Mohamed Bellouki. Amazigh speech recognition using triphone modeling and clustering tree decision, Annals of the University of Craiova 46 (2019), 56–65.
9. A.Boukous, The planning of Standardizing Amazigh language The Moroccan Experience, IR- CAM.
10. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition. Proceedings of the IEEE, (1998), 22782324.
11. A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, 1097-1105. Curran Associates, Inc., (2012).
12. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 11 (2013).
13. O. Parkhi, A. Vedaldi, A. Zisserman, Deep Face Recognition. volume 1 (2015), 41.1-41.12.
14. G. Hu, Y. Yang, D. Yi, J. Kittler, S. Li, T. Hospedales, When Face Recognition Meets with Deep Learning: An Evaluation of Convolutional Neural Networks for Face Recognition, 04 (2015).
15. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-Scale Video Classification with Convolutional Neural Networks,06 (2014), 1725-1732.
16. K. Simonyan, A. Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, Advances in Neural Information Processing Systems, 1- 06 (2014).

17. . Wang, D. J. Wu, A. Coates, A. Y. Ng, End-to-end text recognition with convolutional neural networks. Proceedings of the 21st International Conference on Pattern Recognition, (ICPR2012), (2012), 3304-3308.
18. Y. Kim., Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 08 (2014).
19. Y. LeCun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, D. Henderson, Advances in Neural Information Processing Systems 2. chapitre Handwritten Digit Recognition with a Back-propagation Network. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, (1990), 396-404.
20. https://www.intechopen.com/books
21. Ouhnini, Ahmed & Aksasse, B. & Ouanan, Mohammed. (2023). Towards an Automatic Speech-to-Text Transcription System: Amazigh Language. International Journal of Advanced Computer Science and Applications. 14. 10.14569/IJACSA.2023.0140250.