Analysis, overview and Creation of an Arabic LVCSR

Author: Aratz Puerto

Advisors: Eva Navas, Aitor Alvarez

hap/lap

Hizkuntzaren Azterketa eta Prozesamendua
Language Analysis and Processing

Final Thesis

September 2018

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Language and Communication, Communications Engineer.

Abstract

As the standardized version of the Arabic Language, Modern Standard Arabic (MSA) is the most prevalent form of this language. MSA is also the third most spoken language in the world with over 300 million speakers. Moreover, its history dates back to the eight century B.C, resulting in a strikingly rich linguistic structure. This linguistic structure brings along a broad range of challenges in terms of Large Vocabulary Continuous Speech Recognition (LVCSR) execution. In this dissertation we present an analysis on the Modern Standard Arabic language from a linguistic perspective together with the state of the art of the current Arabic LVCSR from the technical perspective by reproducing and evaluating its state of the art.

Contents

List of Figures

List of Tables

1. CHAPTER

Introduction

1.1 Contextualization

The Modern Standard Arabic, also referred to as MSA, is the official and unified language among all the Arabic countries for culture, media and education from Morocco to the Gulf countries. Far from being a mother tongue, it is usually learned as a second language and serves as a bridge between the aforementioned countries, making it the most widespread variety of the Arabic language.

Since each Arabic region can have several dialects, most people do not speak MSA in a daily basis. Besides, new words are also borrowed and integrated in those dialects, either in their original form or by adaptation to its morphological structure. As a result of its inflectional and agglutinative morphology with gender, number, tense, person and case, a single Arabic word can express a whole English sentence. Said words are created from roots by applying patterns to them.

These facts are but a sample of the morphological challenge MSA poses regarding the application of a Large Vocabulary Continuous Speech Recognition (LVCSR) system. Furthermore, acoustic intricacies are also present in the process, namely the lack of diacritics, which combined with the unvocalized roots of the Arabic language creates a plethora of possible pronunciations for the same word.

The use of the Modern Standard Arabic is on the rise for various reasons, from religion to culture, which makes it appealing for many corporations and the scientific community.

Furthermore, MSA is one of the six official languages of the United Nations and the official language of 28 states, one of the most used languages only preceded by English and French. Vicomtech [1] is an applied research centre specialising in Advanced Interaction technologies, Computer Vision, Data Analytics, Computer Graphics and Language Technologies. As such, as part of the mentioned research facilities, Vicomtech has taken up interest in MSA, applying it to a European project called ASGARD [2] (Analysis System for GAthered Raw Data). In order to enrich the research and application of MSA to the project, this master thesis is also aligned with it.

1.2 Fit in the European project ASGARD

The ASGARD project was born from the need to provide Law Enforcement Agencies (LEAs) with a solution to tackle with the processing of the massive amount of data they employ in a regular basis. This project focuses on building a community between the LEA and Research and Development (R&D) industries by providing and maintaining a toolkit suitable to the LEA needs. Among the integrated technologies, many Speech Processing tools are to be implemented, like Automatic Speech Recognition. Furthermore, these tools are developed in multiple languages, one of them being the Modern Standard Arabic.

1.3 Aim of this master's thesis

The aim of this master's thesis is two fold. On the one hand, by researching its main characteristics, the main features of the Modern Standard Arabic language from the linguistic point of view are examined and outlined. On the other hand, the LVCSR mechanics from a technological perspective are explored in order to create a first version of an MSA LVCSR. To do so, in the next pages, the state of the art of the LVCSR along with the Arabic LVCSR are outlined. The main features and challenges of the Arabic LVCSR are also analyzed. Since, for various reasons, the Arabic dialect is useful inside of the ASGARD project, dialectal Arabic, and specifically Algerian dialect is also looked into among the features of the Arabic LVCSR and the steps to create an Algerian Dialectal LVCSR based on a first version of MSA LVCSR identified. However, the Algeriand Dialectal LVCSR has not been developed. In the second part of this document, the steps followed to gather

[1] http://www.vicomtech.org/
[2] http://www.asgard-project.eu

resources (acoustic and text corpus) and the required preprocessing so as to be able to use them to train a first version of LVCSR are explained. The experiments ran taking the state of the art of the Arabic LVCSR as a starting point as well as the preprocessing needed to do so are also described.

The contents of this dissertation are organized as follows. Chapter 2 gives an overview over the State of the Art of the LVCSR in general (including its evolution, the current LVCSR systems and performance improving methods) and the Arabic LVCSRS in particular. In addition, the features and main challenges of both MSA and Dialectal Arabic are presented. Chapter 3 explains the steps taken to gather and preprocess the resources used to train the first version of our Arabic LVCSR as well as to build the different Acoustic Models (AM), Language Models (LM) and statistical Grapheme-to-Phoneme (G2P). The experiments carried out to recreate the state of the art of the Arabic LVCSR, its evaluation and discussion are included in chapter 4. Conclusions and the future work can be found in chapter 5.

2. CHAPTER

State of the Art

This chapter outlines an overview of the State of the art of the LVCSR systems followed by the current State of the Art of the Arabic LVCSR as well as the challenges it faces. Main characteristics of the Modern Standard Arabic and the Dialectal Arabic are also explained.

2.1 Large Vocabulary Continuous Speech Recognition

With a variety of applications, from dictation to device controlling, the aim of Automatic Speech Recognition (ASR) systems is to transform a speech signal into a sequence of words. There are multiple ASR challenges which mainly differ in the speaking type (read or spontaneous speech), speaker mode (speaker dependent or independent), vocabulary size (small, medium or large), application (isolated or continuous) or background characteristics (clean or noisy), among others.

In this work the focus is put in speaker independent Large Vocabulary Continuous Speech Recognition under any background condition. The aforementioned differences between the ASR challenges define the major difficulties LVCSR systems have to tackle with. Read speech tasks such as dictation are usually easier to carry out than spontaneous speech tasks like the transcription of telephonic speech. Furthermore, speaker independent systems require larger amounts of training data compared to speaker dependent ones in order to face speaker variability. Moreover, large vocabulary systems also have the need to gather enough data to train the acoustic and language models. Finally, while isolated speech

tasks work in single words at a time, continuous speech adds another layer of difficulty in terms of locating word boundaries and the diverse pronunciations related to dialects, co-articulation and noise.

2.1.1 Evolution of the LVCSR

Automatic Speech Recognition has been an extensive field of study for years. Figure 2.1 shows a representation of a typical speech recognition system. Starting from a waveform, firstly feature vectors are extracted. Next, the most likely word sequence for the given feature vectors is estimated using both an acoustic and a language model (Yook, 2003).

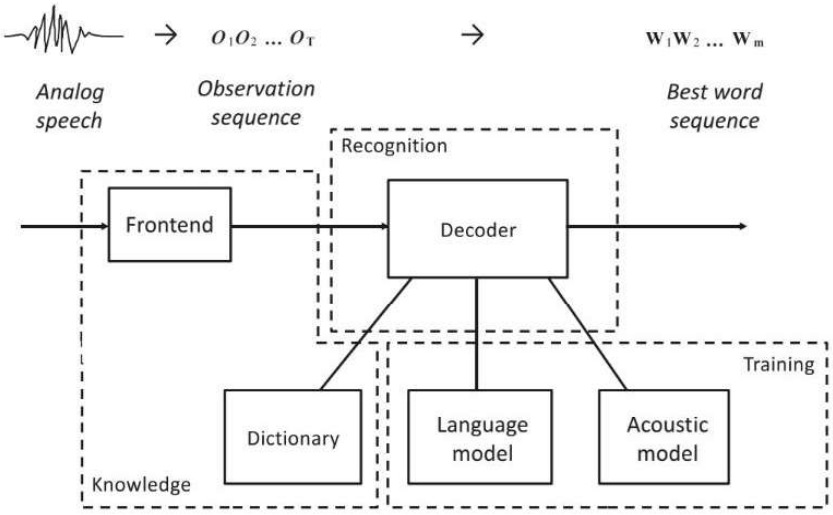

Figure 2.1: Typical Speech Recognition System.

By applying a diversity of techniques over the different architectures of the Acoustic and Language models, such as discriminative training and various adaptation techniques, the results obtained by the LVCSR systems have improved over the years.

Even though the combination of some of these techniques have given good results, the progress achieved by these procedures has been rather slow for years. However, the increase in computing ability along with the development of the deep learning techniques in LVCSR led to a breakthrough in the pace of this advancement in the last years.

In the span of their evolution up to date, the Acoustic Modelling of the LVCSR have gone through a series of architectures. While traditional LVCSR systems were based in HMM-GMM architectures, latest advances in machine learning algorithms and the available

computational power allowed the use of HMM-DNN based LVCSR systems obtaining better results than the former architecture. Nowadays, if enough data is available, end-to-end systems have proven to perform better than the previously mentioned HMM based hybrid architectures.

HMM-GMM based systems

Traditional ASR systems are based in the use of GMMs to represent the relationship between HMM states and the acoustic input.

Given a word sequence W and a speech feature O the aim of ASR is to get the posterior distribution $p(W|O)$. However, obtaining an output sequence composed of discrete symbols (i.e words) based on an input which consists of continuous vectors is not an easy task. Therefore, $p(W|O)$ is rewritten with the Bayes theorem to divide it into the likelihood function $p(O|W)$ (acoustic model) and the prior distribution $p(W)$ (language model) as follows (Watanabe et al., 2017):

$$\widehat{W} = arg\max_{W \in \mathcal{W}} p(O|W)p(W)$$

where:

- $\widehat{W}$: Estimated word sequence.
- $\mathcal{W}$: Set of all possible word sequences.
- $O = \{o_t|t = 1,...,T\}$: T-length sequence of speech feature vectors.
- $W = \{w_n|n = 1,...,N\}$: N-length word sequence.
- $o_t \in \mathbb{R}^D$: D-dimensional speech feature vector at frame t.
- $w_n \in \mathcal{V}$: Word at nth posistion in an utterance with vocabulary $\mathcal{V}$.

This way the search for the best word sequence $\widehat{W}$ on an observation $X = \{x_1, x_2, ..., x_T\}$ is broken down into the elements mentioned above.

Hidden Markov Models, also referred to as HMM are statistical models in which the system being modeled is assumed to be a Markov process with unknown parameters as explained in (Bansal et al., 2008). Since not only each HMM state has a probability distribution over the possible output but the state transition is also probabilistic, the sequence

of tokens generated by an HMM gives information about the sequences of states. This information can be used for speech recognition applications where the role of the HMM is to account for the variability in speech.

Typically an HMM model can be defined as follows: $\lambda = (A, B, \pi)$ (Bansal et al., 2008) where:

- A is the state probability distribution: $A = \{a_{ij}\}$
- B is the observation symbol probability density: $B = \{b_j(k)\}$
- π is the the initial state distribution: $\pi = \{\pi_i\}$

These parameters are estimated in training time for each of the states to get the most probable word by using maximum likelihood estimates of the set of the observations that occur within each one of them.

Gaussian Mixture Models, also known as GMM, are a probabilistic model for representing any distribution of subpopulations within an overall population. GMMs are parametrized by weights, means and covariances (Bansal et al., 2008).

For a GMM with M components, the m^{th} component has a mean of μ_m and a covariance matrix C_m. Based on these components the GMM density is defined as a weighted sum of Gaussian densities:

$$p_g m(x) = \sum_{m=1}^{M} w_m g(x, \mu_m, C_m)$$

where:

- m: Gaussian component (m=1..M)
- M: total number of Gaussian components
- w_m: component probabilities, also referred to as weights
- C_m: covariance matrix
- μ_m: mean
- g: K-dimensional Gaussian probability density function.
  $g(x, \mu_m C_m) = \frac{1}{(2\pi)^{\frac{K}{2}} |C_m|^{\frac{1}{2}}} e^{\frac{-1}{2}(x-\mu_m)^T C_m^{-1}(x-\mu_m)}$

Taking all these components into account, a GMM probability density function is finally defined by a parameter list given by $\theta = \{w_i, \mu_i, C_i\}$ where $i = 1..M$.

While the role of the HMMs is to account for the variability in speech, the GMMs are used to determine how well each state of the HMM corresponds to the coefficients representing the input. In HMM-GMM based ASR systems, the input waveform is typically represented by its Mel-Frequency Cepstral Coefficients (MFCC). Figure 2.2 shows an example of the HMM based phone modeling where it is regarded as a random generator of acoustic vectors consisting of a sequence of states connected by probabilistic transitions to it.

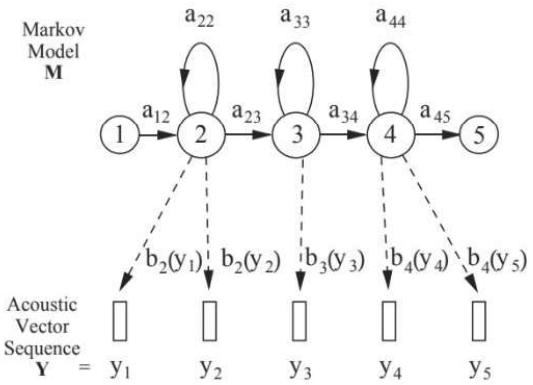

Figure 2.2: HMM-based Phone Model.

HMM-DNN based systems

Advances in both machine learning algorithms and computer hardware led to a shift in the automatic speech recognition paradigm making it possible to train *Deep Neural Networks (DNN)* containing many non-linear hidden units as well as a wide range of output layers which host the HMM. This kind of systems uses DNN networks to produce posterior probabilities over the HMM states instead of making use of GMMs. By using neural networks such as Convolutional Neural Networks (CNN), HMM-DNN systems have shown to outperform HMM-GMM systems on a variety of speech recognition tasks (Abdel-Hamid et al., 2014).

In the case of the DNNs, the representation of $p_g m(x)$ is obtained from a neural network. For a D-dimensional feature vector $o_t \in \mathbb{R}^D$ at frame t in the HMM state j $p_g m(x)$ is decomposed using the Bayes theorem and represented by the frame-level posterior *PDF* $p(j|o_t)$ and represented as follows:

$$p_g m(x) = p(o_t|j) = \frac{p(j|o_t)p(o_t)}{p(j)}$$

where $p(o_t)$ and $p(j)$ are prior distributions of feature vector o_t at HMM state j.

Acoustic models using HMM with likelihoods obtained by DNN are called hybrid HMM-DNN systems and they have proved to outperform conventional GMM-HMM systems in various tasks (Hinton et al., 2012) (Virtanen et al., 2012) Figure 2.3 shows an example of a conventional hybrid HMM-DNN system.

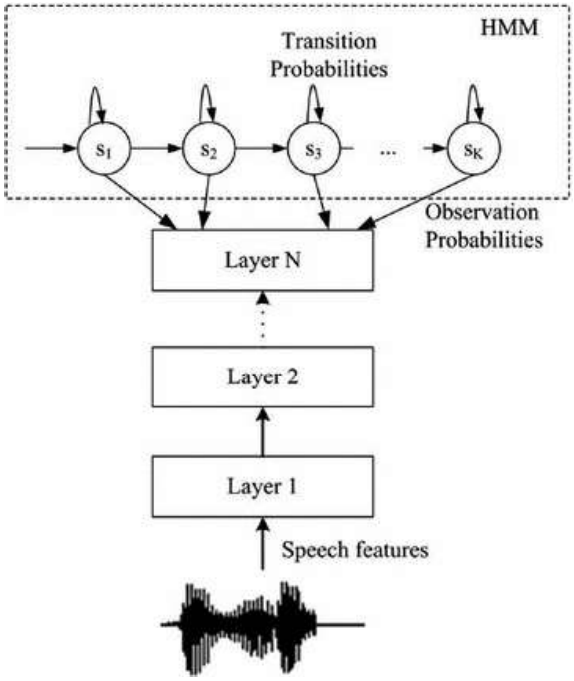

Figure 2.3: Hybrid HMM-DNN system.

Various DNN architechtures have obtained good results in terms of LVCSR. Works like (Hinton et al., 2012) introduce the use of **Feed-forward neural networks**. This kind of DNN proved to perform better with WER rates up to 10% lower. Furthermore, they expose that the spectrogram features of speech work better than MFCCs when fed to a Feed-forward Neural Network compared to their previous use in GMM-HMM systems. A later study by (Graves et al., 2013) explores the use of deep Recurrent Neural Networks (deep RNN) and deep Long Short Term Memory (deep LSTM) by using RNNs to map from an acoustic to a phonetic sequence and using an LSTM architechture to cope with the long range context. Given an unput sequence $z = (x_1, ..., x_T)$, a standard RNN computes the hidden vector sequence $h = (h_1, ..., h_T)$ and output vector sequence $y = (y_1, ..., y_T)$ by iterating the following equations from $t = 1$ to T:

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{hy}h_t + by$$

Where:

- **W**: Weight matrix
- **b**: bias vector.
- **H**: hidden layer function, which is usually an element-wise application of a sigmoid function.
- **x**: input sequence element.
- **y**: output sequence element.
- **h**: hidden vector element.

The LSTM architecture (Hochreiter and Schmidhuber, 1997) contains a set of recurrently connected subnetworks or memory blocks. Each memory block contains memory cells to store the temporal state of the network along with three multiplicative gate units to control the information flow. The input gate controls the information passed from the input activations into the memory cells, while the output gate controls the information passed from the memory cells to the rest of the network. Finally, the forget gate adaptively resets the memory of the cell. This architecture can be used by the LVCSR to handle the long range context intrinsic to the language.

RNN-LSTM is, however, not the only architecture which can be used to model long range dependencies. In (Peddinti et al., 2015) a Time Delay Neural Network (TDNN) architecture is proposed. TDNN has proved to be effective in modelling long range temporal dependencies. The initial transforms are learnt on narrow contexts and the deeper layers process the hidden activations from a wider temporal context. Therefore, the higher layers can learn wider temporal relationships.

In recent years, ***end-to-end*** (E2E) ASR systems have gained popularity. This method simplifies the usual pipeline the previous hybrid systems need to go through since they are based in a pure neural architecture. However, not only are they computationally more expensive but more training data is also required so as to performing as well as DNN-HMM systems. End-to-end systems have proved to perform better than HMM-GMM and HMM-DNN systems and create more robust models which are less sensitive to speaker

variation and noise (Battenberg et al., 2017). Currently E2E systems are divided into two main approaches which differ in how the alignment between observations and output symbols is done and how the dependencies between output symbols are ordered. The former use algorithms like CTC to create explicit alignments. The latter, on the other hand, use encoder-decoder models and do not compute any alignments unless attention mechanisms are used (Watanabe et al., 2017).

Language Model

The language model (LM) aims to determine how probable a word sequence W is ($P(W)$) (Young, 1996). The probability of a word sequence $W = \{w_0, w_1, ..., w_n\}$ can be written as products of conditional probabilities for each word as follows:

$$P(W) = P(w_0, w_1, ..., w_N) = \prod_{i=1}^{N} P(w_i | w_{i-1}, ..., w_1, w_0)$$

Where:

- W: Word sequence
- N: Length of word sequence W.
- w_i: i-th element of sequence W.

n-gram LM

In this work we mainly focus on n-gram based LMs. N-grams provide a simple yet efficient way to achieve this end since it is assumed that w_i only depends on the preceding n-1 words. N-grams are the most used language models where n is usually conditioned by the available training data (Chen and Goodman, 1999). In the case of n-grams the likelihood of the word sequence w_1^n ($P(w_1^n)$) is computed as follows:

$$P(w_1^n) = \prod_{i=1}^{n} P(w_i / w_1^{i-1})$$

Where:

- w_1^n : Target word sequence $(w_1, w_2, ..., w_n)$.

- w_i :i-th word.
- w_i/w_1^{i-1} : Preceding $n-1$ word sequence.

Not only do N-grams focus on local dependencies, making them very effective for languages in which word order is important and the strongest contextual effects tend to come from near neighbours but they also code syntax, semantics and pragmatics at the same time.

N-grams are, however, not the only efficient LMs. There are also various neural network structures that can be used for language modelling such as Feed-forward Neural Network Language Models (FNNLM) (Le et al., 2011), Recurrent Neural Network Language Model (RNNLM) (Mikolov et al., 2010) and Long Short Term Memory based RNNLM (Sundermeyer et al., 2012). RNNLMs deserve especial attention since they have proved to improve the performance (Bengio et al., 2003).

Recurrent Neural Network Language Model

Instead of consisting on the preceding $n-1$ words like in the case n-grams, the input of the RNNLM only consists of the previous word w_{i-1} and a continuous vector v_{i-2} for the remaining context. A connection between the input and hidden layers is added to represent the full history $h_i =< w_{i-1},...,w_1 >$ for word w_i. The input word is coded using one-hot representation in order to obtain its vector representation. The probability of any given sentence W in RNNLMs can be written as follows:

$$P(W) = \prod_{i=1}^{N} P(w_i|w_{i-1},...,w_1,w_0) \approx \prod_{i=1}^{K} P(w_i|w_{i-1},v_{i-2}) \approx \prod_{i=1}^{K} P(w_i|v_{i-1})$$

Where:

- w_i: i-th word in the sequence.
- v_{i-1}: Continuous history vector.
- v_{i-2}: Continuous vector which captures long term history from the start of the sequence via the recurrent connection.

The complete history of word w_i can be represented with two forms. While one makes use of the previous word w_{i-1} and a continuous history vector v_{i-2} the other only uses the

continuous history vector v_{i-1}. Figure 2.4 shows the typical topology of a RNNLM where w_{i-1} along with v_{i-2} are used as input. The hidden layer computes a new history representation v_{i-1} via a sigmoid activation to achieve non-linearity. v_{i-1} is finally passed to the output layer to produce the normalized RNNLM probabilities using a softmax activation.

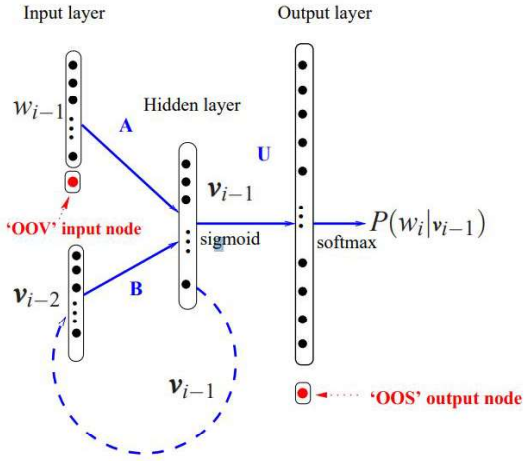

Figure 2.4: Topology of the Recurrent Neural Network Language Model.

Evaluation of the Language Model

In order to evaluate the LM two parameters are taken into consideration, Perplexity (PP) and Out of Vocabulary (OOV) words. PP on a test set is the inverse probability of the test set using a LM, therefore, the lower the PP the better the LM will adjust to this task.

With the aim of evaluating the quality of the language model the perplexity of the test data (of size n) is decomposed as follows for n-gram based LMs:

$$PP = 2^{-\frac{1}{n}log_2(P(w))}$$

The OOV, on the other hand, is expressed in the percentage of unknown words with respect to all the tokens in the test set. Much like PP, the lower the OOV the better.

2.1.2 Performance improving methods

Over the last decade many works have been performed to address the various challenges of the LVCSR systems and improve their performance.

Speaker Adaptation Methods

The aim of the speaker adaptation techniques is to fix the mismatch between the speaker independent parameters of the AM used for the recognition and the characteristics of the target speaker.

Maximum A Posteriori estimation
The Maximum A Posteriori estimation (MAP) adds prior information by using the information about speaker independent features to estimate the speaker specific ones by using the speaker independent models as a prior probability distribution to get the speaker dependent data. Therefore, the estimation of speaker specific models requires less data since it makes use of the information extracted from the prior distribution (Gauvain and Lee, 1994).

The MAP estimate θ_{MAP} is computed as follows:

$$\theta_{MAP} = \underset{\theta}{argmax} g(\theta|x) = \underset{\theta}{argmax} f(x|\theta)g(\theta)$$

Where:

- $x = (x_1, ..., x_T)$: Set of T observation vectors obtained from the probabilistic function of the Markov chain.
- $f(x|\theta)$: Probability density function of variable x.
- $g(\theta)$: Prior distribution for $theta$.
- $g(\theta|x)$ Prior distribution function of parameter θ after observing X (posterior distribution).

By this computation the posterior distribution obtained by a Maximum Likelihood estimation is maximized.

Model-space Maximum Likelihood Linear Regression (MLLR)
Unlike MAP which adapts a model making use of prior information, other techniques estimate a transform to adapt the Gaussian means and covariances so as to maximize the likelihood of the adaptation data (Gales and Woodland, 1996).

The new model mean $\hat{\mu}$ and variance $\hat{\sum}$ are calculated as follows:

- $\hat{\mu} = A\mu + b$

- $\hat{\Sigma} = \Sigma$

Where:

- μ: Speaker independent mean vector.
- $\sum$: Speaker independent variance matrix.
- A: Diagonal transformation matrix.
- b: Bias vector.

Feature-space Maximum Likelihood Linear Regression (fMLLR)

Much like MLLR, fMLLR is a frequently used speaker adaptation approach. However, instead of transforming the Gaussian means μ, fMMLR transforms the model's feature space as follows (Gales and Woodland, 1996):

$$\hat{x^t} = Ax + b$$

Where:

- x^t: Feature vector for time t
- A: Diagonal transformation matrix.
- b: Bias vector.

Vocal Tract Length Normalization (VTLN)

The variability of vocal tract size can lead to a decreased accuracy in terms of LVCSR. VTLN aims to compensate for this speaker dependent inconsistency. Works like (Eide and Gish, 1996) and (Zhan and Waibel, 1997) look into this phenomenon. The variability in vocal tract length is based on a scaling in the frequency axis. Therefore VTLN aims to estimate the warping needed in said axis so as to normalizing the waveform.

Given the frequency f of a signal, the warped signal is computed as follows:

$$\hat{f} = f + arctan(\frac{(1-\alpha)sin(f))}{a-(1-\alpha)cos(f)})$$

Where:

- f: Original frequency of the signal.
- $\hat{f}$: New warped (normalized) frequency of the signal.
- α: Warping factor of the signal.

By this approach the mismatch between the acoustic data and the acoustic model can be fixed.

i-vector based speaker adaptation

While the speaker adaptation approaches mentioned above give good results for GMMs, this procedure is not as clear when it comes to DNNs (Saon et al., 2013). In this approach, instead of adapting the Gaussian mean, identity vectors (i-vectors) are used to estimate the posterior distribution of the feature vectors generated from a Universal Background Model represented as a GMM, and then concatenated to each frame of the acoustic features. This concatenation is the fed to the neural network.

For a given speaker s, the mentioned acoustic feature vectors are represented as K diagonal covariance Gaussians with mixture coefficients c_k, means $\mu_k(s)$ and diagonal covariances ε_k following this distribution:

$$x_t \sim \sum_{k=1}^{K} c_k N(\cdot; \mu_k(s), \varepsilon_k)$$

The speaker data $\{x_t(s)\}$ is first aligned with the GMM to estimate zero-order and centered first-order statistics defined as follows:

$$\gamma_k(s) = \sum_t \gamma_{tk}(s)$$
$$\theta_k(s) = \sum_k \gamma_{tk}(s)(x_t(s) - \mu_k(0))$$

Where:

- $\gamma_k(s)$: Zero-order statistic of mixture component k given $x_t(s)$.
- $\theta_k(s)$: Centered first-order statistic of mixture component k given $x_t(s)$.
- $\gamma_{tk}(s)$: Posterior probability of mixture component k given $x_t(s)$.

The i-vector $w(s)$ is then calculated by estimating the mean of the posterior distribution applying MAP to it. Finally, $w(s)$ is concatenated to every frame $x_t(s)$ to be used as the input of the DNN.

Acoustic data augmentation

Data augmentation is a method to increase the amount of available data for training purposes by generating revamped versions of the original data. Several data augmentation techniques have proven to improve the robustness and help to avoid overfitting when training acoustic models. These techniques include corrupting the data and creating a perturbation in the signal to produce altered versions of the raw data taken as input. Acoustic Data Augmentation is especially useful for DNN based systems, since they need more training data to perform better.

In (Ko et al., 2015) four data augmentation approaches are analyzed.

1. Noise injection: This approach aims to create new audio samples from clean speech by corrupting it with background noise. To do so, given an audio signal x^i and a noise audio signal ξ^i a new noisy speech file can be generated by superposition, getting $\hat{x} = x^i + \xi^i$ as a result as explained in (Hannun et al., 2014).

2. Vocal Tract Length Perturbation (VLTP): Augments the data by modifying the utterances applying a frequency warping factor with a smaller range than the one used in Vocal Tract Length Normalization as in (Jaitly and Hinton, 2013).

3. Tempo perturbation: Tempo perturbation based audio augmentation consists on modifying the speech rate of the original signal by a factor, while ensuring that other intrinsic features of the signal (such as pitch and spectral envelope) are maintained as detailed in works like (Kanda et al., 2013). The variations between the original data and the augmented data require an alignment process for the tempo perturbed utterances.

4. Speed perturbation: This method is carried out by varying the speed of the signal. To do so the sampling rate of the original signal $x(t)$ is modified by a factor α resulting in a new time warped signal $x(\alpha t)$ which is faster or slower than the original depending on the used factor.

All these methods produce altered versions of the initial data, thus, extending the original acoustic corpus. This is especially useful for systems based on deep learning algorithms in which the size of the training corpus has to be big enough in order to achieve good performances.

Lattice Rescoring

Lattice Rescoring is an approach used to add additional information to the LM (Siniscalchi et al., 2009). Once the decoder has created a set of hypotheses, additional information is used to rerank them by a rescoring algorithm.

The lattice expresses the syntactic constraints of the grammar used in training time of the LVCSR system. It is a weighed graph $G(N,A)$ composed of N nodes and A arcs. The nodes contain the timing information and the arcs include recognized symbol along with its score and conveys a word in a hypothesis. The arc scores, W_n, are computed as follows,

$$W_n = \sum_{i=1}^{K} PS_n^i$$

Where:

- PS_n^i: Sum of the logarithm of the phone probabilities of the i-th phone in the n-th arc.
- K: Number of phones in the word related to the n-th arc.

The new scores are then combined with the existing ones to get the new acoustic score S_n as follows,

$$S_n = w_w W_n + w_l L_n$$

Where,

- L_n: Acoustic score before rescoring.
- w_w: Interpolation weights of the word-level score W_n.
- w_l: Interpolation weights of the log-likelihood score L_n.

2.1.3 Current LVCSR solutions

Currently, there are several Automatic Speech Recognition systems, either as commercial use software (such as Voicebase [1], 3PlayMedia [2] or Scribie [3] among others) , or toolkits which allow the creation of such systems (namely HTK (Young et al., 2002), Julius (Lee et al., 2001), Sphinx (Huggins-Daines et al., 2006), RTWH (Rybach et al., 2009), Kaldi (Povey et al., 2011), wav2letter (Collobert et al., 2016), end-to-end (Graves and Jaitly, 2014), Baidu (Battenberg et al., 2017) or deepspeech (Graves and Jaitly, 2014)).

For this master thesis the focus has been put in open source toolkits discarding the commercial systems. Works like (Gaida et al., 2014) give an insight into the advantages and disadvantages of the most popular open source toolkits. Based on this, the use of Kaldi has been favored, since, while computationally more expensive, not only does it give the opportunity to run the most advanced techniques in terms of training and decoding out of the box but it also outperforms its counterparts. Kaldi also features source code and examples for most of the standard techniques, up to the use of deep neural networks. The fact that Kaldi has an exceptional community has also been crucial for choosing this toolkit over the rest.

In order to train the baselines for the experiments carried out in this work, the work in (Ali et al., 2014a) has been followed.

2.2 LVCSR systems for Arabic language

Although there are many commercial Arabic LVCSR systems such as BBN TidesOnTap systems (Billa et al., 2002), IBM ViaVoice[4], Google's STT [5], Votek [6] or Bing Speech [7], most of the current open source Arabic LVCSR, namely LORIA (Menacer et al., 2017) and SRI/Nightingale (Vergyri et al., 2008) are developed using toolkits such as Kaldi.

These works use different approaches to tackle the challenges the Arabic Language poses to develop LVCSR systems for that language.

[1] https://www.voicebase.com/
[2] https://www.3playmedia.com/
[3] https://scribie.com/
[4] https://www-01.ibm.com/software/pervasive/viavoice.html
[5] https://cloud.google.com/speech-to-text/
[6] http://votek.me/
[7] https://azure.microsoft.com/en-us/services/cognitive-services/speech/

2.2.1 Data processing

Special attention is usually put into the data used to train models, especially in terms of text corpora. Since the aforementioned frameworks do not work with Arabic script, a normalization and transliteration phase us usually applied in order to prepare the input for the training of the models.

Normalization

The Arabic language is morphologically lush. For this reason it is not strange to find variations in the different types of text. To cope with this problem a diversity of normalization approaches are commonly used. Some works like (Ali et al., 2014a) include a manual preprocessing stage in which they correct the raw text to mend common Arabic mistakes as well as a semi-manual tagging to detect the spelling mistakes.

Transliteration

In order to be able to train the Language Model a transliteration step is necessary, where Arabic words or lexical items are transliterated into Romanic. This is typically done following the Bukwalter scheme (shown in Figure 2.5) (Habash et al., 2007) which is an ASCII only transliteration scheme, representing Arabic orthography strictly one-to-one, unlike the more common romanization schemes that add morphological information not expressed in Arabic script. This way, for instance, the و waw symbol is transliterated as *w* regardless of whether it is realized as a vowel /u:/ or a consonant /w/.

Arabic letters	ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	ي
Buckwalter	A	b	t	v	j	H	x	d	*	r	z	s	$	S	D	T	Z	E	g	f	q	k	l	m	n	h	w	y
IPA (MSA)	ʔ, aː	b	t	θ	dʒ g ʒ	ħ	x	d	ð	r	z	s	ʃ	sˤ	dˤ	tˤ	ðˤ zˤ	ʕ	ɣ	f	q	k	l	m	n	h	w, uː	j, iː

Figure 2.5: Buckwalter transliteration scheme for Modern Standard Arabic

2.2.2 Acoustic Model

Depending on the chosen architecture the AMs used in the literature for Arabic LVCSR differ significantly.

Hybrid GMM-HMM systems use techniques like Linear Discriminative Analysis (LDA) transformations and Maximum Likelihood Linear Transforms (MLLT) to project the concatenated frames to bigger dimensions trained using different discriminative methods like Maximum Mutual Information (MMI) and Minimum Phone Error (MPE).

In HMM-DNN systems, a variety of procedures are applied in order to get the best performance, such as applying state-level Minimum Bayes Risk (sMBR) criterion and using N-dimensional feature vectors for speaker adaptation (namely feature space Maximum Likelihood Linear Regression (fMLLR)) where fMLLR vectors are used as input layers and the output layers represent the number of HMM states.

Table 2.1 shows the different Acoustic Models and techniques used in different works (such as (Ali et al., 2014a) and (Cardinal et al., 2014)) to obtain the best results of Arabic LVCSR.

	Broadcast	**Conversational**	**Overall**
GMM	22,32 - 28,21	42,62 - 43,53	36,74 - 37,42
GMM + fMLLR	20,98 - 23,65	37,69 - 41,07	32,7 - 34,63
GMM+MPE	19,54	39,07	32,84
GMM+bMMI	19,42	38,88	32,63
SGMM+fMLLR	19,9 - 21,56	**36,05** - 39,08	**30,9** - 32,94
SGMM+bMMI	**18,86**	36,34	30,73
SGMM+fMLLR+MMI	20,9	33,67	29,13
DNN	17,36 - 21,5	34,71 - 35,7	29,81 - 29,85
DNN+MPE	**15,81**	32,21	26,95
DNN+ivector	20,51	34,38	29,44
DNN+fMLLR	20,51	34,03	29,22
DNN+fMLLR+ivector	19,55	32,91	28,16
DNN+fMLLR+MPE	18,93	30,27	26,24
DNN+fMLLR+icev+MPE	17,99	**30,08**	**25,78**

Table 2.1: WER values of several Arabic LVCSR systems using different Acoustic Models and adaptation techniques

Overall the results tend to be better when using DNN over (s)GMMs. The best results which can be observed in the table is 14,81 for broadcast speech using DNN+MPE, 30,08 for conversational speech using DNN+fMLLR+ivectors+MPE and 25,78 for the combined using the same setup.

2.2.3 Language Model

Since Arabic is such a rich language, n-morpheme models are commonly trained. Due to its rich morphology there are many representations of the same root leading to a growth in perplexity due to the inflexional nature of the language (Kirchhoff et al., 2003). In order to train a n-morpheme model the corpus needs to be written in morphemes instead of words following the pattern prefix*-stem-suffix*.

Since the size of the data of a morpheme based LM varies from the one used for n-gram based ones, the perplexity is recomputed as follows (Gauvain et al., 1996):

$$PP_n = 2^{\frac{n_1}{n_2} log_2(PP)}$$

where n_1 is the size of the original data and n_2 the size of the morpheme based one.

2.2.4 Grapheme-to-Phoneme

Grapheme-to-Phoneme (G2P) conversion is the process to convert a written word to its phonetic representation. This relationship is considered transparent for Modern Standard Arabic since the mapping between grapheme and phoneme is one to one (Harrat et al., 2014). This, however, is not that idyllic in the case of the MSA LVCSR systems, due to the fact that vowels and diacritics are lost in the transliteration which creates a high level of ambiguity. For instance, the word كَتَبَ /ktb/ can have different pronunciations, such as /kataba/, /kutiba/, /kutubun/, /kutubi/, /katbin/, etc. In fact, there are 43 different possible pronunciations for this word.

Most of the works on Arabic G2P conversion use two approaches:

1. **Dictionary-based approach:** This method is based on the use of a pronunciation dictionary which contains the correct pronunciation for each word. This way the G2P conversion is reduced to checking the dictionary for the appropriate pronunciation.

2. **Rule-based approach:** In this kind of method phonetic rules are deduced from phonological and phonetic studies of Arabic or learned using a statistical approach in order to do the conversion.

Disambiguation

As explained above, diacritics and vowels are lost in the training of the LM, therefore in order to get as correct a transcription as possible a disambiguation step is added to the pipeline. Toolkits like MADA (Habash et al., 2009) and MADAMIRA (Pasha et al., 2014) give the necessary tools for the disambiguation of the words by selecting the most probable pronunciation among all the possible ones. In the QCRI lexicon[8] a single lexical entrance can have up to 52 different pronunciations with an average of 4 pronunciations per entry.

2.3 Features and Challenges of the Arabic LVCSR

Understanding the particularities of the language is crucial so as to fathom the challenges the Arabic language poses in terms of LVCSR. In this section we explain the main features of the Arabic language (and its dialects) and the challenges it presents.

2.3.1 Main features of the Arabic language

With more than 300 million people speaking it as their first language, Arabic is the largest language in terms of the number of speakers who use it in all its forms. MSA is considered as the formal and unified variety of the language. As such, it is used to reach large audiences, such as broadcast news and newspapers. Native Arabic speakers, however, do not use MSA as their first language in their daily lives. Rather, dialectal (or colloquial) Arabic, which is usually derived from MSA, is commonly used by them in a daily basis as explained in (Ali et al., 2014a). Each region can have one or more dialects which are influenced by the history of the region itself. The dialectal Arabic has been grouped into five regional language groups (Ali et al., 2015): Egyptian (EGY), North African or Maghrebi (NOR), Gulf or Arabian Peninsula (GLF), Levantine (LAV), and Modern Standard Arabic (MSA).

Works like (Menacer et al., 2017) look into the main characteristics of the Arabic features. Not only does MSA boast a complex morphology but it is also characterized by a rich vocabulary which is both inflectional and agglutinative. The grammatical system of the

[8] http://alt.qcri.org//resources/speech/dictionary/ar-ar_lexicon_2014-03-17.txt.bz2

Arabic language is based on root and pattern structure using more than 10000 roots and 900 patterns (Menacer et al., 2017). Thus, Arabic words are derived from root by using said patterns leading to lexical entries that can sometimes correspond to a whole English sentence. New words are borrowed from different languages, namely English, Turkish, Spanish or French and are integrated in the vocabulary of these dialects, some of which are in the original forms and others altered adapting them to the morphological structure of the Arabic language.

In (Droua-Hamdani et al., 2009), the authors mention that the standard Arabic is composed by 28 consonants and 6 vowels: 3 short vowels ([a], [u] and [i]) which are respectively ([fetŸa], [‡amma] and [kasra]), with their 3 opposite long ones ([a:], [u:] and [i:]) as well as ten digits (from 0 to 10). Regarding the alphabet it contains two types of representations (Ali et al., 2014a), characters which are always written and diacritics which are not written in most cases. The optional nature of diacritics also adds to the degree of word ambiguity.

Arabic names and verbs inflect for gender (masculine and feminine), and for number (singular, dual and plural) (Alkuhlani and Habash, 2011). The inflection for gender and number is typically carried out by adding suffixes which bear this information:

- Masculine singular: +∅
- Feminine singular: ة+ +$\hbar^2$
- Masculine dual: ان+ +An
- Feminine dual: تان+ +tAn
- Masculine plural: ون+ +wn
- Feminine plural: ات+ +At

This pattern is not followed for broken plurals, which changes the noun's structure varying their case/state forms. In these cases the form of the singular suffix is inconsistent with the word's number, which is plural. For instance, the word كاتب (writer/scribe) hast two broken plurals, كتاب ktAb (masculine) and كتبة ktb$\hbar$ (feminine).

Furthermore, Arabic also has a class broken feminine in which the feminine singular form is derived. For example, the adjective 'red' أحمر has the two following form/function

pairs: أحمر ÂHmr (masculine singular/masculine singular) and حمراء Hmra' (masculine singular/feminine singular). Adding to this, some irregular form/function also exist as well as non-countable collective plurals that behave as singulars even though they may translate to other languages as plurals.

This inconsistency, however, is not present in the case of nominal duals and verbs.

The relationship in some constructions, such as nouns and their adjectives and verbs and their subjects is more complex in Arabic than it is in other languages in terms of morpho-syntactic agreement. In the case of the adjectives, except for the non-human plural irrational, which always takes feminine singular adjectives, the Arabic adjectives agree with the nouns they modify in gender and number. As for the verbs and their nominal subjects, they follow the same rules as the adjectives with the exception of the verb-subject order which only agrees in gender to singular number. Furthermore, while numbers over 10 always take a singular noun, numbers 3 to 10 take a plural one and inversely agree with the noun's functional gender. Figure 2.6 shows an example of these 3 morpho-syntactic agreement cases.

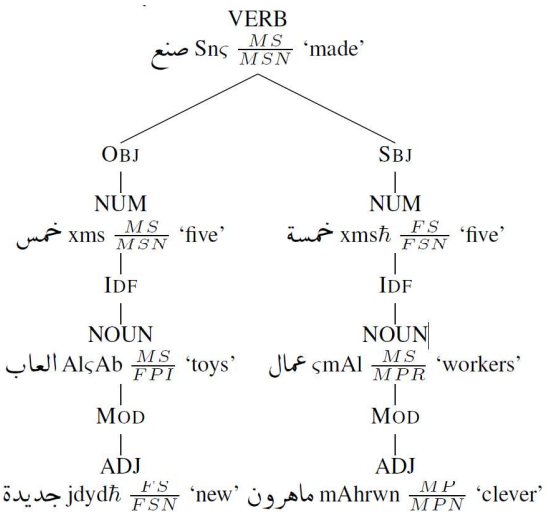

Figure 2.6: Arabic morpho-syntactic agreement example.

2.3.2 Dialectal Arabic

Arabic dialects are often classified regionally, as Egyptian, North African, Levantine Gulf, Yemeni or sub-regionally, namely, Tunisian, Algerian and Lebanese.

Although, as explained above, MSA is the official and unified language among all the

Arabic countries, it is usually learned as a second language. Native speakers seldom use it in spontaneous discourse. In cases where spoken MSA would be required, the speakers draw upon code switching between their native dialect and MSA (Abu-Melhim, 1991).

While Dialectal Arabic (DA) and MSA have some common features, such as their rich inflectional morphological complexity, they also differ in other aspects.

Phonology

A clear example of phonological discrepancy is the pronunciation of the words which contain the letter ق q (Qaf) in their MSA counterpart. In Tunisian Arabic, the pronunciation of this consonant is /q/. However, while in Egyptian and Levantine Arabic it is /’/ (glottal stop) in Gulf Arabic it is /g/ (Haeri, 1991).

Orthography

As a unified language, MSA has an established standard orthography. However, this is not the case for DA. People often write words either matching their phonology or the etymology of these words. Nevertheless, some works like (Habash et al., 2012) present a proposal for a standardized dialectal orthography (COD: Conventional Orthography for Dialectal Arabic).

Morphology

Morphological differences are quite common. One example is the future marker particle which appears as +س sa+ or سَوْ sawfa in MSA. It appears as +ح Ha+ or رح raH in Levantine dialects, +هــ ha+ in Egyptian and باش bAš in Tunisian (Bouamor et al., 2014).

Lexicon

The lexical variation between MSA and DA is quite considerable and it is by far the biggest of the biggest discrepancy between both types of Arabic (Ibrahim, 2008).

Syntax

Opposed to the significant lexical differences, syntactic ones are minor. For instance, the negation can be conveyed as ما mA, مـشْ mish, لا lA, لن lam, etc. However, overall the syntactic distribution of both DA and MSA is uniform to a large extent among varieties (Benmamoun, 2011).

2.3.3 Algerian Dialect vs. MSA

Situated in the north of Africa, Algeria extends over the vast territory of 2,380,000 km^2 occupied by about 34.8 million inhabitants whose majority are concentrated in the north of the country. The official language of the Algerians is SA, but their mother tongues are either Tamazight (Berber language) or specific variants of SA language which are stemming from the ethnic, geographical and colonial influences (Droua-Hamdani et al., 2009).

The efforts made, until now, to develop ASR for Arabic dialects concern those considered as close to MSA, namely Iraqi, Egyptian, Qatari and Levantine (Menacer et al., 2017). However, there are few ASR systems for Maghrebi dialects especially those used in Algeria. However, due to the lack of Arabic dialect data not much work has been done in these grounds compared to that of MSA. The regular contact of the Algerian speakers with several languages allowed them a freedom of choice for code-speaking where the speakers diversify their communication strategies using sometimes a language, sometimes another language namely, French, Berber and Turkish. Also many speakers use two languages at the same time, referred to as code-switching, and sometimes glide from one language to another. Many researchers believe that this transition phenomenon occurs particularly when the speaker is unable to tell the right word in the language. Algerian dialect (ARZ) is very different from Arabic dialects of the Middle-East, since it is highly influenced by the French language. The differences between ARZ and MSA are very extensive (Habash et al., 2013). As mentioned before, the lexical differences are very significant, for instance, the counterpart of the ARZ word المَاكْلَ (food) in MSA would be طعام. Phonology also differs between these forms of Arabic. For instance, the MSA consonant ث /θ/ is pronounced as /t/ in ARZ (or /s/ in more recent borrowings of MSA) (Habash et al., 2012). In terms of morphology, not only are there morphemes in ARZ that do not exist in MSA such as the negation ش+ +ما mA + +š, but there are also MSA features that are not present in ARZ, most notably case and mood. Among the morphemes that exist in both,

there is a change in the morpheme form, i.e., the MSA future marker +س sa+ appears in ARZ as +هـ ha+ (Habash et al., 2013). Orthographic differences follow the same pattern as the rest of the dialectal Arabic having different writing rules.

2.3.4 Main challenges

Even though classic techniques for ASR systems can be efficiently applied to Arabic speech recognition, it is necessary to take into account language specificities to improve the system performance.

As explained above, Arabic language is characterized by a complex morphological structure where different kinds of prefixes and suffixes are appended to the word stems producing a very large number of inflectional forms. This leads to poor LM probability estimates and thus high LM perplexities causing problems in large vocabulary continuous speech recognition (LVCSR). This explains the high out-of-vocabulary (OOV) rate compared with English language which consequently leads to the increase of the Word Error Rate (WER) (Habash, 2010). These are, however, not the only points which need assessment, some of which are specific to the language. This is why Natural Language Processing (NLP) applications committed to Arabic (such as MADA) are vital at the preprocessing step before calculating the language model.

Furthermore, the absence of diacritics in Arabic texts is a serious issue for many applications in NLP (Al-Anzi and AbuZeina, 2017). For every Arabic root which is not vocalized, the ASR system has to consider all the possibilities of pronunciations or has to restore the diacritics. However, since the continuous speech naturally has some acoustic variations that are not accounted for in the pronunciation dictionary, it is almost impossible to consider all possible variants in the pronunciation which leads to increased error rates due to the mismatch between the acoustic features of the speech signal and the phonetic transcription. Therefore, it is important that the phonemes of the pronunciation dictionary are representative of the actual contents of the training data.

Finally, the difficulty to obtain corpora for dialects that are spoken rather than written adds up to the challenges of the Arabic LVCSR (Al-Anzi and AbuZeina, 2017).

Main challenges of ARZ

Even though MSA and ARZ are related, there are many phonological, morphological and lexical differences between them. While the majority of the tools and resources developed for Arabic NLP are devoted to MSA they are scarce in the case of ARZ which is a spoken language without conventional writing rules. Even though there are proposals for the standardization of its orthography (such as CODA) the scientific community has yet to adopt a unified writing system for ARZ or any of the versions of DA. Furthermore, ARZ lacks resources such as corpora for this dialect. Moreover, without a standardized writing system lexicons and G2P are also scarce for ARZ.

Some works like (Harrat et al., 2014) have confronted these challenges by creating their own resources. To do so, they first set up a writing system by checking if there is a MSA word close to the dialect word. If such a word exists, they use the MSA version of the word, otherwise writing it as it is spoken. Once the writing system is fixed the corpus is manually transcribed by following it. ARZ follows the same rules as MSA in terms of G2P conversion. However, ARZ boasts a variety of borrowed words from foreign languages (especially French) by either altering the word phonologically or with the same pronunciation as in the source language. Since the former is an Arabic version (in terms of phonology) it follows the rules of the Arabic language. The latter, on the other hand, does not follow said rules making it more difficult the G2P conversion. Therefore, some French phonemes must be included in the ARZ phone set. The transcriptions are then converted into phonemes after a diacritic restoration phase (toolkits such as ADAD (Harrat et al., 2013) are typically used to do so). Finally using the newly created lexicon a rule based G2P is trained (either statistical or rule based one).

3. CHAPTER

Experimental Setup

In this section we explain the steps followed for the creation of the AM, LM and G2P and lexicon needed to build our LVCSR for the Arabic Language including the preprocessing of the corpora. Concerning the AM, the preparation carried out in the acoustic corpus, the features of the trained AM and the techniques used to improve their performance are described. Regarding the LM, the gathering of additional text to enrich the text and its transliteration are outlined. The different lexicons used and the trained statistical G2P are also explained.

3.1 Data processing

In this section we present the steps followed to prepare both the acoustic and text corpora for the experiments presented in Chapter 4.

3.1.1 Acoustic Corpus

The GALE Arabic Phase 3[1] [2] [3] corpus was developed by the Linguistic Data Consortium (LDC) and features approximately 128 hours of Arabic broadcast conversation and 260 hours of Arabic broadcast news for a total of 388 hours of Arabic speech collected by

[1] https://catalog.ldc.upenn.edu/LDC2016S01
[2] https://catalog.ldc.upenn.edu/LDC2016T17
[3] https://catalog.ldc.upenn.edu/LDC2017S02

LDC, MediaNet, Tunis, Tunisia and MTC, Rabat, Morocco during Phase 3 of the DARPA GALE (Global Autonomous Language Exploitation) Program. This acoustic corpus has been used for the training of the different AMs in the experiments explained in Chapter 4.

GALE Arabic Phase3 is composed of three parts. While part1 and part3 boast broadcast news speech collected in 2007, part 2 contains broadcast conversation speech collected between 2007 and 2008. Regarding their content, the broadcast news recordings in part1 and part3 feature news broadcasts focusing principally on current events and the broadcast conversation recordings in part2 contain interviews, call-in programs and round-table discussions focusing principally on current events

All the audio files in the corpus are provided in FLAC format, 16 kHz, 16 bits and 1 channel.

Acoustic Corpus Preprocessing

In order to prepare the audio files for the training of the different LVCSR systems built in this work, some acoustic preprocessing has been applied to the acoustic corpus.

1. **Audio conversion:** Since the audio files are provided in FLAC format, they are first converted into wav format using *avconv* [4] for this end.

2. **Split by utterance:** All the audio files in the acoustic corpus have a corresponding Tab Delimited Format (TDF) file with the information about the audio itself. TDF is a simple file format in which data is represented as a set of records which are in turn a set of fields separated by tab characters.

   The TDF format for LDC transcripts is a set of 13-field records plus some meta-information. This format was originally designed for use with LDC's new transcription tool XTrans. The 13-field record is also called segment, and all segments in the file are identical. Table 3.1 shows the mentioned 13 fields.

   By applying the acoustic data augmentation process explained in section 2.1.3 to transform the speed and pitch of the audio files, making them faster or slower by a random factor, an alignment process is needed in order to update fields 3 and 4 of the TDF file. To simplify this process the audio files from the acoustic corpus have been split by utterance. Since the start and end time of each utterance can

[4] https://libav.org/avconv.html

Number	Field	Description	Type
1	file	file name or id	string
2	channel	audio channel	number
3	start	start time	number
4	end	end time	number
5	speaker	speaker name or id	string
6	speakerType	speaker type	string
7	speakerDialect	speaker dialect	string
8	transcript	transcript	string
9	section	section id	number
10	turn	turn id	number
11	segment	segment id	number
12	sectionType	section type	string
13	suType	SU type [5]	string

Table 3.1: TDF fields per audio file

be retrieved from the TDF fields, SoX [6] have been used to split the audio files into smaller ones, creating files which only contain speech (and occasional noise). This way the corpus has been split into 114 Conversational speech hours and 204 Broadcast News hours to a total of 318 hours.

3. **Filter by language:** Some parts of the acoustic corpus, such as interviews in the broadcast part, contain non MSA languages (namely English, French and non-MSA Arabic Dialects). These audio files, along with those which contain noise between uttered sentences, have been filtered out of the corpus resulting in a filtered acoustic corpus of 317,61 hours. Regarding ARZ, its vocabulary is influenced by MSA among other languages. ARZ mainly borrows words from the French language either phonologically altered or pronounced exactly in French. Therefore, this step should be different in the case of ARZ since the French words should not be filtered out of the acoustic corpus.

4. **Acoustic Data Augmentation:** The purely MSA audio files have been augmented by using the Speed based acoustic data augmentation method explained in 2.1.3. The factor used to modify the sampling rate of the signal is randomly chosen between 0.9 and 1.1 per file, creating an extra augmented version of each audio file, increasing the total duration of the corpus to 635 hours (206 conversational + 429 broadcast news hours). By calculating the duration of each independent augmented audio and replacing the end time in the corresponding field for time codes in the

[6] http://sox.sourceforge.net/

original TDF file containing the various information about the original not augmented file, we avoid the need to run a more complex new alignment process.

3.1.2 Text Corpus

The text corpus used to train the baseline only contained the transcriptions of the acoustic corpus. This corpus boasts 9,523,547 Arabic words. In order to further enrich the Language Model, various Arabic web newspapers have been crawled with the goal of retrieving the text in their articles. In addition, text obtained from the AraCorpus[7] has also been added.

Web Crawling

Since the target of this work is MSA we chose to scrap newspapers written in MSA (discarding pieces of news like opinion columns) in order to get as similar a text as the one we were using. A total of 8 online newspapers were crawled (7356 articles) for a total of 3,378,580 words. Table 3.2 shows the information about the crawled newspapers, number of articles, number of crawled words and the year range of the articles.

Newspaper	Source	#articles	#words	Years
Al-Ahram	http://www.ahram.org.eg/	339	153,805	2014 - 2018
Al Wafd	https://alwafd.news/	340	67,252	2017 - 2018
Al-Hayat	http://www.alhayat.com/	3,473	2,063,641	2015 - 2018
Elkhabar	http://www.elkhabar.com/	268	109,969	2017 - 2018
Alwasat	http://www.alwasatnews.com/	634	146,681	2015 - 2018
Al Bayan	http://www.albayan.ae/	2,003	729,627	2016 - 2018
BBC Arabic in Arabic	http://www.bbc.com/arabic	13	6,690	2018
CNN Arabic	http://arabic.cnn.com/	283	100,915	2017 - 2018
Total		**7,353**	**6,757,160**	

Table 3.2: Crawled data information

The initial text corpus containing only the transcriptions of the acoustic corpus has been extended this way by 6,757,160 more words after the web crawling process. With the addition of the text from AraCorpus the new text corpus' size is increased to 80,645,615 words.

[7] http://aracorpus.e3rab.com/index.php?content=english

Normalization

The Arabic languages' morphology is extremely broad, therefore, it is not unlikely to find inconsistencies in written text. In order to normalize the crawled text we have first cleaned all the *html* entities from the crawled data as well as any ASCII character (except the numbers) and then follow the same approach as in (Ali et al., 2014a) using MADA to auto-correct the raw input text. This way, non Arabic numbers, several especial characters such as Hamza and diacritics are normalized resulting in a consistent and purely MSA corpus.

The MADA+TOKAN Toolkit
MADA [8] is a freely available toolkit for Arabic NLP applications. Given a raw Arabic text, it adds as much lexical and morphological information as possible by disambiguating in one operation part-of-speech (POS) tags, lexemes, diacritizations and full morphological analyses. TOKAN is a tool which can produce a tokenization formatted to the user's needs from MADA's output.

MADA is divided into 5 sub-components which can be executed as a pipeline or as standalone operations:

1. **Preprocessing:** The preprocessing component can take raw text (one-sentence-per-line), clean it, add foreign word tags, insert whitespace between punctuation and words, and convert UTF8 to Buckwalter.

2. **Morphoanalysis:** it generate, for each input word, a list of possible analyses, with no regard to context.

3. **Generate SVM + ngram files:** it determines N-gram statistics for diacritic word forms and lexemes, and creates back-off lexicons for the next step.

4. **SVMTools:** it runs an independent SVM classifier for a number of MADA features, determining a prediction for that feature value for each word.

5. **Select Morphoanalysis:** For each word, it examines each of the possible analyses and scores each one. The score is developed by comparing the features of each analysis to the SVM prediction; analyses that have agreement with the prediction are given a weighted increase in score. Some additional, non-SVM features are factored in as well. The scores are then normalized, sorted and labeled. Tie-breaking

[8] https://academiccommons.columbia.edu/doi/10.7916/D86D60BS

is employed to guarantee that only one analysis for each word is designated as the correct one.

Even though MADA+TOKAN is a powerful toolkit combination, the requirements needed for them to work may make the setup daunting. Furthermore, the use of MADA has been limited in this work due to errors when setting up one of the dependencies. Therefore, we have not been able to use the whole MADA pipeline and, as a result, MADA has only been used for text normalization.

Transliteration

In this work, transliteration has been done following the Buckwalter scheme explained in section 2.2.2 to get the Romanic representation of the Arabic words both in the text corpora and the lexicon. Since in written MSA vowels and diacritics are not included, MADA tools have been used to do the normalization and include them in the Romanic representation as shown in Figure 2.5.

Table 3.3 shows an example of the Romanic representation obtained from a sentence in Arabic script using the Buckwalter scheme:

Arabic Script	Romanic Representation
غير صريحة بل بالعكس فخبذا إذا	w>msy ElY Aldktwr l>nh fy |xr
الإجابة أه مبهمة وغير أه يعني	AstDAfp >h TrHt Elyh Als&Al Al<jAbp
استضافة أه طرحت عليه السؤال	>h mbhmp wgyr >h yEny
وأمسي على الدكتور لأنه في آخر	gyr SryHp bl bAlEks fHb∗A < ∗A
عليه سؤال جدا غاية في الأهمية	Ely w>TrH Elyh s&Al jdA gAyp
علي وأطرح	fy Al>hmyp

Table 3.3: Example of the Romanic representation of Arabic script.

3.2 Acoustic Modelling

Different types of Acoustic Models have been trained during the experiments, including HMM-GMM and hybrid HMM-DNN systems (both TDNN and RNN+LSTM). The models are trained with the standard 13 dimensional cepstral mean-variance normalized

(CMVN) Mel-Frequency Cesptral Coefficients (MFCC) features without energy, and its first and second derivatives. For each frame, we also include its neighboring ± 4 frames and apply Linear Discriminative Analysis (LDA) transformation to project the concatenated frames to 40 dimensions, followed by Maximum Likelihood Linear Transform (MLLT).

We use this setting of feature extraction for all the models trained in our system. Speaker adaptation is also applied with feature-space Maximum Likelihood Linear Regression (fMLLR). Our system includes all conventional models supported by KALDI: diagonal Gaussian Mixture Models (GMM) and DNN models. Training techniques such as MLLT discriminative training are also employed to obtain the best performance. MFCC features are extracted from speech frames and MFCC+LDA+MLLT are then used to train the Speaker Independent (SI) GMM model. fMLLR are estimated based on each training utterance with SI GMM. Furthermore, fMLLR transformed features are then used for DNN training separately.

In the end, we obtain two different sets of models: GMM-HMM based models and DNN-HMM based models. The system will use the intermediate basic GMM model for first pass decode to obtain fMLLR transformation, and the second pass decoding with one of the more advanced final models. These models are all standard 3-states context dependent triphone models. The GMM-HMM model has about 1K Gaussians per state over 8K states. The DNN-HMM model is trained with 3 layers. Each layer has 2K nodes. The learning rate is 0.0003. Furthermore, since it is not a bidirectional LSTM, only chunk left context of 40 have been used.

The baseline has been created using the 317,61 hours of the original corpus after splitting it into utterances and filtering foreign languages. Further AMs have been trained by also adding the 317,61 hours of the augmented data to the previous corpus for a total of 635,22 hours.

It is worth noticing that the acoustic modelling for ARZ has to be different from the MSA since in ARZ French words are used. The MSA phonemes need to be extended with the French phonemes used in ARZ shown in Figure 3.1.

3.3 Language Modelling

We trained the LM using the KALDI system (kaldi_lm) which produces lattices as recognition results. Various approaches have been used in order to train different LMs. During

Phoneme	Dialect word	Pronunciation	English gloss
/ɛ/ /e/	مَاسْتَار	/m a s t ɛ R/	master degree
/g/	بَاقَاج	/b a g a ʒ/	luggage
/p/	پُورْتَابْل	/p ɔ r t a b l/	cell phone
/v/	پُوفْوَار	/p u v w a r/	power
/y/	سِكُورِتِي	/s e k y r i t e/	security
/ɑ̃/	سُورْمُون	/s y R m ɑ̃/	surely
/ø/ /œ/ /ə/	شُومُور	/ʃ ɔ m œ r/	unemployed
/ɛ̃/	لْكُوزَا	/l k u z ɛ̃/	the cousin
/ɔ̃/	كُونْتْر	/k ɔ̃ n t r/	against
/ʒ/	جِيسْت	/ʒ y s t/	right

Figure 3.1: The French phonemes used in Algerian dialect with an example of dialectal word for each phoneme.

training 3-grams have been computed. Besides, LM perplexity was small enough to skip the pruning process. The same setup has been used at decoding phase. However, 5-grams have been computed, again without pruning for re-scoring purposes. We used the transcriptions for the original corpus after splitting it by utterance and filtering the foreign languages and using the grapheme QCRI lexicon to train the baseline.

Further LMs have been trained using the QCRI pronunciation lexicon with 2,022,705 lexical entries and the statistical G2P explained below in section 3.4.

Language Model Adaptation and Rescoring

In order to adapt the LM to the new text corpus composed by the scraped data and the AraCorpus a LM Adaptation process has been applied. A new lexicon is created so that it contains all the words in the corpus. Their phonetic representation has been created following the same approach as the lexicon used to train the LM that is to be adapted.

For the QCRI lexicon a 1:1 mapping between characters and phonetic representation has been applied. Regarding the G2P lexicon the trained G2P model has been used to get the representation of each word in the lexicon. Finally, for the QCRI pronunciation lexicon, due to the lack of linguistic knowledge we could not get the phonetic representations for the new words. Therefore that model was not adapted.

Based on the lexicon, the new phone set is generated so as to represent all the pronunciations in the new LM. A new LM is then trained with this data using unprunned 2-grams. During LM Adaptation KenLM (Heafield, 2011) has been used instead of kaldi_lm with

Kneser-Ney modified smoothing. Finally a new graph is created which is then decoded using the same acoustic model which represents the phone set of the target LM.

Much like during training, in order to rescore the new lattice arc scores, 5-grams are built which will be the base of the new rescoring models.

In terms of DA, while MSA has strict linguistic rules and a typographic writing system which are followed when writing, ARZ lacks any standard or rules when it comes to writing words which affects the estimation of the probabilities of the language model.

In (Menacer et al., 2017) they make use of the PADIC [9] corpus to cope with this problem by applying the following rules when writing in ARZ: if a dialectal word does exist in MSA, it must be written such as in MSA, otherwise the word is written as it is pronounced.

Test data

A total of 3 hours have been selected from the GALE acoustic corpus for testing purposes composed by 2 hours of broadcast news and 1 of conversational speech. The selected testing data have also been filtered following the process explained in *Chapter 2, section 2.2.1: Transliteration*. The transcriptions of the 3 hours of the acoustic corpus left out for testing purposes have been used as test text corpus. These transcriptions have a total of 74,006 words.

3.4 Dictionary based G2P

Due to the lack of linguistic knowledge of the language we have trained a statistical G2P converter making use of the QCRI pronunciation lexicon using the data-driven grapheme-to-phoneme converter SEQUITUR (Bisani and Ney, 2008). Furthermore, following this approach the out of vocabulary word problem is also tackled by choosing the most probable pronunciation for words which are not present in the lexicon.

The last 1000 lexical items with a frequency of 1 have been used as testing data, leaving 2,021,707 entries for training. A total of 11 models have been trained with the rates of incorrectly transcribed phonemes or Symbol Error Rates (SER) shown in Table 3.4. Each of the models is used as the base for the following model in an iterative way. The SER is computed as follows:

[9] https://sourceforge.net/projects/padic/

$$SER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}$$ Where:

- **S**: Number of substitutions.
- **D**: Number of deletions.
- **I**: Number of insertions.
- **C**: Nmber of correct symbols.
- **N**; Number of symbols in the reference (N=S+D+C)

Model	SER
model-1	28,56
model-2	32,55
model-3	26,33
model-4	21,12
model-5	18,02
model-6	13,4
model-7	10,2
model-8	**9,8**
model-9	9,84
model-10	9,8
model-11	9,78

Table 3.4: Symbol Error Rates of the trained statistical G2P.

Since the SER of the models start stabilizing at *model-8* with an error rate of 9,8% we have decided to use that model of the G2P in this work.

3.5 Lexicon

Three different lexicon have been used in this work:

- **QCRI lexicon:** The QCRI lexicon[10] boasts 526K unique grapheme words collected from a news archive from many news websites as well as the Arabic news website Aljazeera.net. It is processed using MADA tools and the least frequent words have been discarded in order to create this lexicon. There is a 1:1 mapping between

[10] http://alt.qcri.org//resources/speech/dictionary/ar-ar_grapheme_lexicon_2016-02-09.bz2

each symbol in the lexical entry and the phonetic representation of said symbol. Therefore, there is only one pronunciation per lexical entry.

- **QCRI pronunciation lexicon:** The QCRI pronunciation lexicon (Ali et al., 2014b) is based on the QCRI lexicon and contains different pronunciations for each of the lexical entries with an average of 3.84 pronunciations for each grapheme word for a total of 2 million possible pronunciations.

- **Extended QCRI pronunciation lexicon:** An extended lexicon has been produced using the QCRI pronunciation lexicon and adding the most frequent words in the new gathered corpus. Since we had no way to reproduce the phonetic representation of the lexical items which appeared in the QCRI pronunciation lexicon for the new words, the trained G2P has been used to get the most probable pronunciation for said new entries. This extended lexicon boasts 2,100,588 lexicon entries.

4. CHAPTER

Experimentation and Evaluation

4.1 Description

In this chapter the different experiments carried out in this work are explained along with the obtained results. The obtained results are then discussed. Different types of AM and LMs have been trained. Different lexicons have also been used in the process, including the training of a dictionary based statistical G2P.

4.2 Experiments and results

The experiments carried out are based on the work in (Ali et al., 2014a). The aim of our work is to reproduce and possibly improve the results in it in order to create a first version of an Arabic LVCSR by using the GALE Arabic recipe [1]. Table 4.1 shows the results obtained by Ali et al. for the various AM and LMs they tried. In this work the use of DNN proves to be more effective in terms of AM modelling for the Arabic language. These results improve even more when applying MPE as the criterion for discriminative training. Thus, the best WER obtained by this system is 15,81% for broadcast news speech and 32,21% for conversational news speech (26,95% for the combined speech).

However, the recipe above mentioned only trains some of the GMM and DNNs in Table

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/gale_arabic/s5b

	Broadcast	Conversational	Overall
GMM	22,32	43,53	36,74
GMM+fMLLR	20,98	41,07	34,63
GMM+MPE	19,54	39,07	32,84
GMM+bMMI	19,42	38,88	32,63
SGMM+fMLLR	19,9	39,08	32,94
SGMM+bMMI	**18,86**	**36,34**	**30,73**
DNN	17,36	35,7	29,81
DNN+MPE	**15,81**	**32,21**	**26,95**

Table 4.1: WER results of the related work for each AM

4.1. Table 4.2 shows the WER for the GMM and DNNs in common in both the related work and the recipe.

	Broadcast	Conversational	Overall
GMM	22,32	43,53	36,74
GMM+fMLLR	20,98	41,07	34,63
DNN	**17,36**	**35,7**	**29,81**

Table 4.2: WER results of the related work for each AM (GALE Arabic)

As mentioned above, DNNs perform better than GMMs in all cases. Furthermore, since the recipe does not apply MPE as discriminative training criterion, the best WER is 17,36% for broadcast news speech and 35,7% for conversational news speech (29,81% for the combined speech).

4.2.1 Baseline

The first system built is a baseline system that will be used as a reference. In this system the AM has been trained using the 318 hours of filtered GALE acoustic corpus, 203 hours being broadcast news speech and the other 115 broadcast conversational speech. The LM used in this experiment has been trained using the transcriptions from said audio files, with an unpruned 3-gram, making use of the QCRI lexicon using the one-to-one mapping approach of the Buckwalter representation without taking into account the different pronunciations each lexical item might have. Table 4.3 shows the results obtained using the configuration explained above.

In our experiments we follow the typical pipeline to train train different GMMs which are then used as a base to train the DNNs. First the monophone system is trained. Then, from

the monophone model first and second triphone systems are trained *(tri1, tri2a,tri2b)*. The second triphone *(tri2b)* is the equivalent of the *GMM* in the related work, which includes LDA feature estimation and MLLT discriminative training. The last phase of GMM training is the training of the third triphone pass *(tri3b)*, which, in addition to LDA+MLLT as the previous pass, also includes SAT. The alignment of the third triphone pass includes fMLLR adaptation making it the counterpart of *GMM+fMLLR* in the related work.

Two types of DNNs are trained in the baseline from the GMMs: A RNN+LSTM network and a TDNN which are the equivalent of the DNN in the related work.

Broadcast		**Conversational**	
tri1	43,40	tri1	58,50
tri2a	42,27	tri2a	56,69
tri2b	38,52	tri2b	53,26
tri3b	35,61	tri3b	49,78
RNN+LSTM	**20,50**	RNN+LSTM	**33,24**
TDNN	23,85	TDNN	36,72
Combined (Broadcast x Conversational)			
tri1	48,81		
tri2a	47,40		
tri2b	43,73		
tri3b	40,62		
RNN+LSTM	**25,02**		
TDNN	28,40		

Table 4.3: Baseline System Word Error Rates

Since the corpora used for the related work and our experiments is not the same a mismatch in the results could be expected. Not only are both corpora different phases of the GALE corpus, but while the related work is composed of 284 hours of speech our baseline is trained with 318 hours. These differences create a mismatch between the WER in our baseline and the ones in the related work. Regarding the DNN, however, while TDNN does not outpeform the related work (except for the case of the combined speech in 1,41%), the RNN+LSTM performs better than the related work in the case of the conversational news speech (2,46% improved WER) and the combined speech (4,79%) while it performs 3,14% worse for the broadcast news speech. Since this is not common, we assume that the broadcast news speech in the corpus we used must be more complex than the one in the reference.

4.2.2 Data Augmentation

A second set of AMs have been trained adding speed perturbed data to the 318 hours of the baseline. Due to computational restraints, a 1-fold data augmentation has been applied, therefore doubling the corpus to 636 hours (406 broadcast news hours + 230 broadcast conversational speech hours). The speed is perturbed by a random factor between $(0.9, 1.1)$. The LM used in this experiment is the same as the one used in the baseline. To test the impact DA has on the AM an RNN+LSTM has been trained since it is the best performing DNN in the baseline. Table 4.4 shows the results for the application of the data augmentation experiment.

Broadcast		**Conversational**	
tri1	45,62	tri1	60,41
tri2a	44,92	tri2a	58,86
tri2b	44,92	tri2b	55,26
tri3b	37,79	tri3b	51,92
RNN+LSTM	**20,89**	RNN+LSTM	**33,55**
Combined (Broadcast x Conversational)			
tri1	50,85		
tri2a	49,9		
tri2b	46,05		
tri3b	42,79		
RNN+LSTM	**25,37**		

Table 4.4: Data Augmentation Word Error Rates

The results obtained in this experiment are very similar to the baseline. Therefore it can be concluded that, either not enough data has been augmented to get a significant decrease in the WER or the speed perturbation has not given any useful information for this purpose. The GMM obtain a worse WER of 4,29% in average in the case of the broadcast news speech and 2,07% for the conversational speech news (2,23 for the combined speech news). However, the RNN+LSTM, while slightly worse, perform almost as well as the unaugmented system with less than 0,4% WER in each of the cases.

4.2.3 LM Adaptation

Even though three different lexicons have been introduced in this work, only two of them have been used in terms of LM Adaptation as explained in Chapter 3:

- **QCRI+NEWS:** LM adapted by using the QCRI lexicon generalized to the NEWS domain by using the scraped corpus + AraCorpus.
- **PL+G2P:** LM adapted by using the lexicon created from the statistical G2P + QCRI pronunciation lexicon generalized to the NEWS domain by using the scraped corpus + AraCorpus.

In this experiment the LM used in the baseline has been extended using the scraped corpus and the AraCorpus, using 3-grams for the creation of the adapted LM and 5-grams for rescoring purposes. Since LSTM has outperformed the rest of the AMs, we put the focus on this AM when adapting the LM. Table 4.5 shows the results obtained with this experiment.

Broadcast		**Conversational**	
QCRI+NEWS	16,22	QCRI+NEWS	31,06
QCRI+NEWS+Rescoring	**13,56**	QCRI+NEWS+Rescoring	**25,68**
PL+G2P	22,97	PL+G2P	37,60
PL+G2P+Rescoring	19,30	PL+G2P+Rescoring	32,90
Combined (Broadcast x Conversational)			
QCRI+NEWS	19,94		
QCRI+NEWS+Rescoring	**16,32**		
PL+G2P	26,52		
PL+G2P+Rescoring	22,35		

Table 4.5: LM Adaptation for baseline Word Error Rates

Both the QCRI lexicon and the Pronunciation Lexicon (along with the trained statistical G2P) have been used to adapt the LM in this experiment. A significant WER reduction can be observed in the two kinds of speech in the case of the QCRI+NEWS, obtaining decreases of 4.28% in the case of broadcast news speech and 2,18% in conversational news speech (5,08% for the combined speech news). By applying lattice rescoring the performance is increased, with WER around 3,8% in average and obtaining a reduction of of 6,94%, and 5,56% in broadcast and conversational (as well as a 8,7% reduction in the case of combined speech) respectively.

Unlike QCRI+NEWS, PL+G2P does not outperform the baseline. The results, produce higher WERs than the ones obtained in the baseline, with increased WERs of 2,47% for broadcast news speech and 4,36% (1,5%). This might be related to the ambiguity created by the multiple pronunciations of each lexical item in the pronunciation lexicon as well as the fact that we were not able to use MADA tools to disambiguate them. Nevertheless,

lattice rescoring obtains decreased WER in the case of PL+G2P too, with a reduction on the WER of 1,2% for the broadcast news speech and 0,34% for the conversational broadcast news (2,67% for the combined speech news).

Data Augmentation + LM Adaptation

The same experiment has been carried out for the data augmented set of AMs. Since, in the case of the LM adaptation of the baseline the best results have been obtained used LSTM as AM and adapting the LM using the QCRI+NEWS, we focus on this setup for this experiment. Table 4.6 shows the obtained results.

Broadcast		**Conversational**	
QCRI+NEWS	16,15	QCRI+NEWS	30,75
QCRI+NEWS+Rescoring	**13,75**	QCRI+NEWS+Rescoring	**25,17**
Combined (Broadcast x Conversational)			
QCRI+NEWS	19,82		
QCRI+NEWS+Rescoring	**16,30**		

Table 4.6: LM Adaptation for data augmented AM Word Error Rates

Since the results of both sets of acoustic models were very similar, it does not strike as surprising that applying the same LM adaptation to this set of AM produces similar WER as the ones obtained adapting the baseline. The results, however, are in general slightly better, with a reduction of 0,07% in the case of the broadcast news speech and 0,31% for the conversational news speech (0,12% for the combined news speech) without applying lattice rescoring. Lattice rescoring decreases the WER even further in almost all the cases. The WER is increased by 0,19% for the broadcast news speech but decreased by 0,51% for conversational news speech (a reduction of 0,02% is also obtained for the combined speech news).

4.3 Discussion

In the experiments outlined in this section we have been able to reproduce and improve the results of the baseline by 6,75% for broadcast news speech and 8,07% for conversational news speech (8,72% for the combined speech news) down to a total of 13,75% WER for broadcast news speech and 25,17% for conversational news speech (16,30% for combined speech news).

Regarding the acoustic models, different kinds of GMMs have been trained which have then been used as a base to train two kinds of DNNs, RNN+LSTM and TDNN. In general, HMM-DNN hybrids outperfom the classical HMM-GMM system. Moreover, among them RNN+LSTM obtain better results than the TDNN. This might be due to the fact that while the TDNN might fail to capture the temporal locations of the long range dependencies, RNN+LSTM generalize better by capturing this information and learning its dependencies thanks to their memory feature.

The different performance improving methods (LDA feature estimation, MLLT discriminative training and fMLLR speaker adaptation) used during the training phase have also proved useful by helping reduce the WER in each AM trained. However, we have not been able to prove the value of data augmentation in the used corpus for the Arabic language either due to lack of a higher order of augmentation or lack of ability to add any further information from the augmented data.

Regarding the LM, adaptation by the QCRI lexicon has outperformed the one done with the pronunciation lexicon (+statistical G2P). This might be related to the inability to use MADA tools for the disambiguation of the proper pronunciation of the different lexical items in the pronunciation lexicon thus increasing its perplexity. However, LM adaptation has improved the performance of the system in all cases, especially when applying lattice rescoring, in an average of 7% over the different types of speech for a WER of 13,75% in broadcast news speech, 25,17% for conversational news speech and 16,30% for the combined speech.

5. CHAPTER

Conclusions

5.1 Conclusions

In this work LVCSR systems in general, and Arabic LVCSR (both MSA and dialectal) have been studied with the aim to be able to replicate a chosen related work and improve it while creating a first version of an Arabic LVCSR. All in all that aim has been achieved by not only replicating the results in the related work but also improving them. The steps needed to adapt this first MSA LVCSR to the Algerian dialect have also been identified.

One of the biggest problems of the Arabic LVCSR systems is the lack of useful resources which meet the needs of the system itself. For this end especial attention has been put into gathering and preprocessing both the acoustic and text corpus up in order to get as high a quality corpus as possible. The acoustic corpus have been split by speaker and audio files with noise or non MSA speech sections have been removed in order to prevent noises and foreign languages from affecting the final quality of the LVCSR. Speed based data augmentation has also been applied so as to doubling the size of the acoustic corpus (in hours) by creating revamped versions of the original corpus with different speed and pitches. Regarding the text corpus different newspapers written in MSA have been scrapped from the web and added to the AraCorpus. This new text corpus has then been normalized using MADA tools and transliterated into Romanic using the Buckwalter representation in order to be able to use Kaldi to train the first version of an Arabic LVCSR. This normalization phase cleans the text of any unwanted character, such as numbers and non-Arabic characters, but it also recovers missing information in Arabic script since in

written Arabic vowels and diacritics are not written but they are pronounced. While we have experimented with three lexicons, two lexicons have been used in this work. The QCRI lexicon, which maps a Buckwalter character to a phoneme on a 1:1 representation and the QCRI pronunciation lexicon which boasts multiple entrances for each possible pronunciation of a written Arabic word. From the QCRI pronunciation lexicon a statistical G2P has also been trained to check the impact it might have on the LVCSR. However, as a result of the strict requirements of MADA Tools, we have not been able to use its full potential due to errors when installing some dependencies. Therefore, we have not been able to use it to disambiguate the different pronunciations of the QCRI pronunciation lexicon which has affected the performance of the trained statistical G2P.

Regarding the AM, different models have been trained. Various HMM-GMM models have been used as a base to train two different HMM-DNN hybrid systems with memory in order to capture the long range dependencies intrinsic to the language. A RNN-LSTM and a TDNN have been trained. RNN-LSTM systems have outperformed any other AM in this work. The use of feature estimation, discriminative training and speaker adaptation have also proved to improve the performance of the AMs, thus obtaining WERs of 20,89% for broadcast news speech and 33,55% for conversational news speech (25,37% for the combined news speech). While these values are not exactly the same as in the related work they are similar enough to pin it down to the fact that different corpora has been used in both works. However, the WERs obtained by our system are 3,53% higher for broadcast news speech and 2,15% lower for conversational news speech (4,44% lower in the case of the combined news speech).

In order to try and further increase the performance of our LVCSR some work has also been done in terms of LM. The WER explained above have been obtained using 3-gram LMs. The LMs themselves were small enough not to need any kind of pruning. By making use of two of the three lexicons explained above (the phonetic representation of the lexical items of the pronunciation lexicon could not be reproduced) along with the gathered and preprocessed text corpus, two different LM adaptations have been performed. On the one hand the QCRI lexicon has been used along with the text corpus to adapt the LM to the news domain. On the other hand, by using the pronunciation lexicon (and the trained statistical G2P to compute the phonetic representation of the OOV lexical items in the text corpus) a second adaptation to the news domain has been performed. Both adapted LMs have significantly improved the performance of our first Arabic LVCSR down to WERs of 13,75% for broadcast news speech and 25,17% for conversational news speech (16,30% for the combined news speech).

5.2 Future work

Even though the aim of the work has been achieved, there is still future work left to do in order to improve this first version of the Arabic LVCSR and adapt it to the ARZ.

Further AMs can be trained as well as performance improving methods to check their effect on the Arabic LVCSR. While they are in the related work, subspace GMM have not been trained since the recipe was not prepared to do so. MPE and MMI have not been applied to either the GMM or the DNN. Additional AMs include combination of the HMM-DNNs as well as the use of new ones, such as the Bidirectional LSTMs including BLSTM, TDNN-LSTM and TDNN-BLSTM. Since E2E systems have proved to perform better than the hybrid HMM-DNN LVCSR systems, this kind of AM should also be experimented with.

As stated above the focus of the work in this dissertation has been put in the technical aspect. Works like (Cardinal et al., 2014) carry out intensive linguistic preprocessing in order to obtain more suited LM for the language. Different types of LMs should also be looked into. For instance, Works like (Khurana and Ali, 2016) show the advantages of using RNN-LMs for the Arabic language. Since works like (Choueiter et al., 2006) show the advantages of using n-morpheme based LMs for Arabic, this kind of LMs should also be included in future works.

During the span of this work MADA tools could not be used in its totality. We believe being able to complete the process would give us important linguistic information to be able to create better LMs, thus further improving the performance.

5.2.1 Dialectal Arabic

Through this work we have identified the different actions which need to be taken in order to train an ARZ LVCSR system based on the trained first version of the MSA LVCSR. Since, acoustically, ARZ is mainly affected by the French language, when filtering out foreign (non-MSA) audio files, the ones containing the French language should be kept. Regarding AM, audios with French speech are now kept in the acoustic corpus, the French phonemes shown in Figure 3.1 need to be modelled in the acoustic model. Ideally, a transcribed corpus of spoken ARZ would be used for this matter. However, such a resource, much like the text corpus, is not available. Works like (Menacer et al., 2017) propose training all the acoustic models corresponding to the 31 French phonemes and

then adding them to the 34 original MSA models. Moreover, concerning the LM, the lack of resources as well as an standard in terms of dialectal scripting make it impossible to gather text resources. For this reason, corpora like PADIC [1] which fix a set of rules to cope with the problem just described needs to be used.

[1] https://sourceforge.net/projects/padic/

Appendices

Bibliography

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

Abu-Melhim, A.-R. (1991). Code-switching and linguistic accommodation in arabic. In *perspectives on Arabic linguistics III: papers from the Third Annual Symposium on Arabic Linguistics*, volume 80, pages 231–250. John Benjamins Publishing.

Al-Anzi, F. S. and AbuZeina, D. (2017). The impact of phonological rules on arabic speech recognition. *International Journal of Speech Technology*, 20(3):715–723.

Ali, A., Dehak, N., Cardinal, P., Khurana, S., Yella, S. H., Glass, J., Bell, P., and Renals, S. (2015). Automatic dialect detection in arabic broadcast speech. *arXiv preprint arXiv:1509.06928*.

Ali, A., Zhang, Y., Cardinal, P., Dahak, N., Vogel, S., and Glass, J. (2014a). A complete kaldi recipe for building arabic speech recognition systems. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 525–529. IEEE.

Ali, A., Zhang, Y., Cardinal, P., Dahak, N., Vogel, S., and Glass, J. (2014b). A complete kaldi recipe for building arabic speech recognition systems. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 525–529.

Alkuhlani, S. and Habash, N. (2011). A corpus for modeling morpho-syntactic agreement in arabic: gender, number and rationality. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 357–362. Association for Computational Linguistics.

Bansal, P., Kant, A., Kumar, S., Sharda, A., and Gupta, S. (2008). Improved hybrid model of hmm/gmm for speech recognition.

Battenberg, E., Chen, J., Child, R., Coates, A., Gaur, Y., Li, Y., Liu, H., Satheesh, S., Seetapun, D., Sriram, A., et al. (2017). Exploring neural transducers for end-to-end speech recognition. *arXiv preprint arXiv:1707.07413*.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Benmamoun, E. (2011). Agreement and cliticization in arabic varieties from diachronic and synchronic perspectives. *al-'Arabiyya*, pages 137–150.

Billa, J., Noamany, M., Srivastava, A., Liu, D., Stone, R., Xu, J., Makhoul, J., and Kubala, F. (2002). Audio indexing of arabic broadcast news. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 1, pages I–5. IEEE.

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451.

Bouamor, H., Habash, N., and Oflazer, K. (2014). A multidialectal parallel corpus of arabic. In *LREC*, pages 1240–1245.

Cardinal, P., Ali, A., Dehak, N., Zhang, Y., Hanai, T. A., Zhang, Y., Glass, J. R., and Vogel, S. (2014). Recent advances in asr applied to an arabic transcription system for al-jazeera. In *Fifteenth annual conference of the international speech communication association*.

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Choueiter, G., Povey, D., Chen, S. F., and Zweig, G. (2006). Morpheme-based language modeling for arabic lvcsr. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I. IEEE.

Collobert, R., Puhrsch, C., and Synnaeve, G. (2016). Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*.

Droua-Hamdani, G., Boudraa, M., and Selouani, S. (2009). Algerian arabic speech database project (ALGASD): Corpora's elaboration. In *3rd International Conference on Arabic Language Processing (CITALA'09)*. Citeseer.

Eide, E. and Gish, H. (1996). A parametric approach to vocal tract length normalization. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 1, pages 346–348. IEEE.

Gaida, C., Lange, P., Petrick, R., Proba, P., Malatawy, A., and Suendermann-Oeft, D. (2014). Comparing open-source speech recognition toolkits. *Tech. Rep., DHBW Stuttgart*.

Gales, M. J. and Woodland, P. C. (1996). Mean and variance adaptation within the mllr framework. *Computer Speech & Language*, 10(4):249–264.

Gauvain, J., Lamel, L., Adda, G., and Matrouf, D. (1996). The limsi 1995 hub3 system. In *Proc. DARPA Speech Recognition Workshop*, pages 105–111.

Gauvain, J.-L. and Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE transactions on speech and audio processing*, 2(2):291–298.

Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Habash, N., Diab, M. T., and Rambow, O. (2012). Conventional orthography for dialectal arabic. In *LREC*, pages 711–718.

Habash, N., Rambow, O., and Roth, R. (2009). Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, volume 41, page 62.

Habash, N., Roth, R., Rambow, O., Eskander, R., and Tomeh, N. (2013). Morphological analysis and disambiguation for dialectal arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 426–432.

Habash, N., Soudi, A., and Buckwalter, T. (2007). On arabic transliteration. In *Arabic computational morphology*, pages 15–22. Springer.

Habash, N. Y. (2010). Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Haeri, N. (1991). Sociolinguistic variation in cairene arabic: Palatalization and the {\it qaf\/} in the speech of men and women.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Harrat, S., Abbas, M., Meftouh, K., and Smaili, K. (2013). Diacritics restoration for arabic dialects. In *INTERSPEECH 2013-14th Annual Conference of the International Speech Communication Association*.

Harrat, S., Meftouh, K., Abbas, M., and Smaïli, K. (2014). Grapheme to phoneme conversion-an arabic dialect case. In *Spoken Language Technologies for Under-resourced Languages*.

Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huggins-Daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., and Rudnicky, A. I. (2006). Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I. IEEE.

Ibrahim, Z. (2008). Lexical variation: Modern standard arabic. *Encyclopedia of Arabic Language and Linguistics*, 3:13–21.

Jaitly, N. and Hinton, G. E. (2013). Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117.

Kanda, N., Takeda, R., and Obuchi, Y. (2013). Elastic spectral distortion for low resource speech recognition with deep neural networks. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 309–314. IEEE.

Khurana, S. and Ali, A. (2016). Qcri advanced transcription system (qats) for the arabic multi-dialect broadcast media recognition: Mgb-2 challenge. In *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pages 292–298. IEEE.

Kirchhoff, K., Bilmes, J., Das, S., Duta, N., Egan, M., Ji, G., He, F., Henderson, J., Liu, D., Noamany, M., et al. (2003). Novel approaches to arabic speech recognition: report from the 2002 johns-hopkins summer workshop. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–I. IEEE.

Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. (2015). Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Le, H.-S., Oparin, I., Allauzen, A., Gauvain, J.-L., and Yvon, F. (2011). Structured output layer neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5524–5527. IEEE.

Lee, A., Kawahara, T., and Shikano, K. (2001). Julius—an open source real-time large vocabulary recognition engine.

Menacer, M. A., Mella, O., Fohr, D., Jouvet, D., Langlois, D., and Smaïli, K. (2017). Development of the arabic loria automatic speech recognition system (alasr) and its evaluation for algerian dialect. *Procedia Computer Science*, 117:81–88.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *LREC*, volume 14, pages 1094–1101.

Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Rybach, D., Gollan, C., Heigold, G., Hoffmeister, B., Lööf, J., Schlüter, R., and Ney, H. (2009). The rwth aachen university open source speech recognition system. In *Tenth Annual Conference of the International Speech Communication Association*.

Saon, G., Soltau, H., Nahamoo, D., and Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. In *ASRU*, pages 55–59.

Siniscalchi, S. M., Svendsen, T., and Lee, C.-H. (2009). A phonetic feature based lattice rescoring approach to lvcsr. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 3865–3868. IEEE.

Sundermeyer, M., Schlüter, R., and Ney, H. (2012). Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Vergyri, D., Mandal, A., Wang, W., Stolcke, A., Zheng, J., Graciarena, M., Rybach, D., Gollan, C., Schlüter, R., Kirchhoff, K., et al. (2008). Development of the sri/nightingale arabic asr system. In *Ninth Annual Conference of the International Speech Communication Association*.

Virtanen, T., Singh, R., and Raj, B. (2012). *Techniques for noise robustness in automatic speech recognition*. John Wiley & Sons.

Watanabe, S., Delcroix, M., Metze, F., and Hershey, J. R. (2017). New era for robust speech recognition.

Yook, D. (2003). Introduction to automatic speech recognition. *Department of computer science, Korea University*.

Young, S. (1996). A review of large-vocabulary continuous-speech. *IEEE signal processing magazine*, 13(5):45.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al. (2002). The htk book. *Cambridge university engineering department*, 3:175.

Zhan, P. and Waibel, A. (1997). Vocal tract length normalization for large vocabulary continuous speech recognition. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.