Context Recognition Methods using Audio Signals for Human-Machine Interaction

by

Mohit Shah

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2015 by the
Graduate Supervisory Committee:

Chaitali Chakrabarti, Co-Chair
Andreas Spanias, Co-Chair
Visar Berisha
Pavan Turaga

ARIZONA STATE UNIVERSITY

May 2015

ABSTRACT

Audio signals, such as speech and ambient sounds convey rich information pertaining to a user's activity, mood or intent. Enabling machines to understand this contextual information is necessary to bridge the gap in human-machine interaction. This is challenging due to its subjective nature, hence, requiring sophisticated techniques. This dissertation presents a set of computational methods, that generalize well across different conditions, for speech-based applications involving emotion recognition and keyword detection, and ambient sounds-based applications such as lifelogging.

The expression and perception of emotions varies across speakers and cultures, thus, determining features and classification methods that generalize well to different conditions is strongly desired. A latent topic models-based method is proposed to learn supra-segmental features from low-level acoustic descriptors. The derived features outperform state-of-the-art approaches over multiple databases. Cross-corpus studies are conducted to determine the ability of these features to generalize well across different databases. The proposed method is also applied to derive features from facial expressions; a multi-modal fusion overcomes the deficiencies of a speech-only approach and further improves the recognition performance.

Besides affecting the acoustic properties of speech, emotions have a strong influence over speech articulation kinematics. A learning approach, which constrains a classifier trained over acoustic descriptors, to also model articulatory data is proposed here. This method requires articulatory information only during the training stage, thus overcoming the challenges inherent to large-scale data collection, while simultaneously exploiting the correlations between articulation kinematics and acoustic descriptors to improve the accuracy of emotion recognition systems.

Identifying context from ambient sounds in a lifelogging scenario requires feature extraction, segmentation and annotation techniques capable of efficiently handling long duration audio recordings; a complete framework for such applications is presented. The performance is evaluated on real-world data and accompanied by a prototypical Android-based user interface.

The proposed methods are also assessed in terms of computation and implementation complexity. Software and field programmable gate array based implementations are considered for emotion recognition, while virtual platforms are used to model the complexities of lifelogging. The derived metrics are used to determine the feasibility of these methods for applications requiring real-time capabilities and low power consumption.

To my parents

ACKNOWLEDGEMENTS

This thesis would have not been possible without the support of my advisors, Dr. Andreas Spanias and Dr. Chaitali Chakrabarti. I am grateful to them for providing me with this opportunity and resources to pursue my research interests and constantly motivating me to perform better. Their guidance on technical subjects has contributed immensely towards this thesis.

I am grateful to Dr. Visar Berisha for providing valuable feedback and innovative ideas related to my research. I would like to thank Dr. Pavan Turaga for his valuable suggestions. I would like to extend my sincere thanks to Brian Mears, with whom I have had the opportunity to collaborate on several occasions over the past few years. Our brainstorming sessions were always enjoyable and gave me an opportunity to learn many valuable lessons.

I would like to extend my gratitude to Prasanna Sattigeri for being a great friend, roommate, colleague. Our many discussions over coffee and video games have, no doubt, had a positive impact on my research. Special thanks to my friends for their unconditional support throughout my studies.

I am really grateful to my wife, Tania, for supporting me unconditionally. She has stood by me, through good and bad times, while living on a different continent at times. Finally, I would like to thank my parents and my sister for always wishing the best for me and encouraging me to freely pursue my interests.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

The creation of artificially intelligent machines or agents is one of the most actively studied and difficult problems of the past fifty years. A machine is considered truly intelligent if humans are unable to discern whether they are interacting with a real human being or a machine based on the responses provided by the latter, i.e. a Turing test [1]. This problem is far from being solved completely, yet, this pursuit has received a tremendous boost in recent years, mainly due to - (i) growth in availability of portable devices, and (ii) advances in computational methods for data analysis. The first has led to a widespread use of smartphones, tablets or wearables, each equipped with a plethora of sensors for data collection on a large scale. The second has demonstrated the usefulness of machine learning algorithms and probabilistic modeling frameworks towards the extraction and analysis of patterns which are relevant to human beings in their daily interactions.

The gap in human-machine interaction has narrowed significantly; commercial applications show that machines are quite capable of engaging humans to a considerable extent. For instance, applications such as Siri and Google Voice respond to phrases or questions uttered by human beings. Similarly, advances in computer vision have enabled machines to identify what human beings see through image-based object and face recognition techniques. A majority of the research is devoted to understand *what* is being said or seen as opposed to *how* it is said or seen. Understanding the latter is quite difficult owing to its subjective nature, yet, it is very essential towards creating artificially intelligent machines.

1.1 Context-Aware Systems

This process of understanding *how*, or context recognition, typically requires machines to extract and analyze the current situation or circumstances surrounding human beings. It is worthwhile to consider a few scenarios to understand the importance of such information. A few examples are provided below - Call centers routinely employ automated voice response systems to deal with customers. These systems can be augmented by identifying whether the customer is angry or nervous based their voice characteristics, information that can be used to promptly alert the supervisor. A smartphone can automatically decide to switch to silent mode if it learns that the user is in a meeting, or alternatively turn up the volume in crowded areas like restaurants and markets. Wearables, such as Google Glass [2], or autonomous vehicles and robots can be configured to provide detailed information and recommendations based on the user's environment. Numerous prototypes of context-aware devices and applications have been demonstrated over the last few decades [3, 4, 5, 6, 7]. Various sensors such as accelerometers, temperature sensors, microphones and touch sensors were used in these systems.

This dissertation focuses on context recognition using audio signals, such as speech and ambient sounds. These signals convey rich information pertaining to a user's plans, behavior and environment, which is often not available (occlusions in images) or difficult to capture from other sources (location or movement). Interaction between humans and machines is considered under different scenarios and applications - (i) *active* interaction between humans and machines, central to emotion recognition and keyword detection, and (ii) *passive* interaction, where the machine assumes the role of an always-on, passive listener, as observed in a lifelogging application. Computational methods, based on machine learning and probabilistic frameworks, are presented for learning robust features and classifiers. Performance evaluations are accompanied by hardware/software implementations, wherever applicable, in order to determine their feasibility for real-time recognition and low-power consumption.

1.2 Supra-Segmental Features for Emotion Recognition

Emotions constitute a fundamental aspect of human-human communication. They either motivate human actions and decision-making, or enrich the meaning of human communication. Traditional speech interfaces, ignore a speaker's emotions, consequently, ignoring highly important information available in the interaction process. Such interactions are frequently perceived as cold, incompetent, and socially inept. Human-centered interfaces must have the ability to detect subtle changes in the speakers' behaviors, especially related to their emotional state, and to appropriately modify their responses. The mapping of an utterance to an emotion is a multi-stage process, involving the extraction of acoustic, low-level descriptors (LLD), representation, and classification. The expression and perception of emotions is highly subjective and varies across languages, and cultures, hence, determining the appropriate set of features and representation methods that generalize well across such conditions is considered quite challenging [8].

Previously, studies have found high-level, supra-segmental representations [9, 10, 11, 12] extracted from low-level descriptors (LLD) to be more successful than segmental methods based on hidden Markov models (HMM) [13, 14]. Typically, such features are extracted by performing a brute-force collection of statistics over LLDs. Yet, this approach does not offer a generative explanation of how emotions affect the observed acoustic characteristics of speech. Furthermore, these features cannot discriminate well between emotions with similar arousal patterns, such as happy-angry or neutral-sad.

1.2.1 Contributions

In this dissertation, a novel feature extraction method using latent topic models (LTM) [15, 16] is proposed. Originally intended for categorization of text documents based on their underlying topics, LTMs are extended to learn supra-segmental features from emotional speech. Topics, in this case, capture emotionally salient information from the the co-occurrence behavior of LLDs. The proposed approach offers a generative model-based explanation of how emotions influence the observed acoustic characteristics of speech, while, overcoming the need for popular, brute-force based feature extraction methods. Experiments are performed over multiple databases with different languages, accents and varying emotion expressions. In each case, the proposed features outperform existing state-of-the-art features. The performance gains are even significant for valence-based classification and longer duration turns, both, considered challenging tasks in the field of acoustic emotion recognition.

The key contributions are as follows:

- An unsupervised learning method based on replicated softmax models (RSM) is extended to a supervised model, leading to more discriminative features and improved performance.

- A point-wise mutual information-based measure is proposed to qualitatively assess the relationship between the derived features, emotions, and low-level acoustic descriptors.

- Cross-corpus studies are conducted to assess the generalization ability of these features. The bias specific to each database due to their respective annotation procedures is identified. Two strategies, instance selection and weight regularization, are proposed to eliminate this bias and improve performance.

- The proposed feature extraction methodology is extended to multiple modalities including facial expressions and spoken content. Individually, the 3 modalities are shown to perform best at recognizing sadness (speech), happiness (face) and neutral (language) emotions, while their combination retains these properties and improves the performance significantly.

- An FPGA-based implementation of LDA is presented; a parallel architecture is devised, which provides speed-up by a factor of 200 over optimized software implementations, while simultaneously satisfying real-time constraints.

- The software implementation complexity of a multi-modal emotion recognition system is assessed to determine its feasibility for real-time applications; a turn of 1 s duration can be processed in 666.65 ms.

The work related to this topic is reported in [17, 18, 19].

1.3 Articulation Constrained Learning

In addition to the acoustic properties of speech, it is commonly understood that speech articulation also exhibits a strong correlation with the emotional state of a speaker. The kinematics associated with tongue, jaw or lips are modulated according to the emotion. Hence, methods that combine both acoustic and articulation descriptors would potentially yield more accurate and reliable emotion recognition systems.

There are relatively very few attempts to characterize emotions using articulatory information. In [20], it was shown that the degree of jaw opening increased significantly as subjects became annoyed (or irritated), while, in [21], the lateral lip distance between the corners of the mouth was shown to be strongly influenced by the emotional state. In [22], the authors showed that articulation-based features achieved a much better classification rate compared to acoustic features for a single male subject. These studies are mostly limited to single subjects, or multiple speakers recorded under similar conditions. However, more importantly, these methods require articulatory data to be available during the recognition step in order to perform reliably. Acquisition of such data on a large scale is difficult and time-consuming due to its invasive and highly sensitive recording procedure, which limits the scope and application of these methods to only laboratory environments.

1.3.1 Contributions

In this dissertation, an articulation constrained learning approach, that requires articulatory data only during training, is proposed to overcome the aforementioned limitations. Specifically, a traditional, L1-regularized logistic regression cost function is extended to include constraints that enforce articulatory reconstruction. Thus, information from acoustic features and articulatory data is jointly captured, which is expected to lead to improved emotion recognition. Multiple databases, providing articulatory information in the form of electromagnetic sensors attached to a speaker's tongue, jaw and lip, or in the form of motion capture markers located at various points on a speaker's face, are used for evaluating this method. Results show a significant improvement for both, speaker-dependent and speaker-independent, emotion recognition tasks on peripheral vowels including */AA/*, */AE/*, */IY/* and */UW/*.

The key contributions are as follows:

- A constrained learning method, that requires articulatory data only during training and generalizes well to unseen samples, is presented.

- A single objective function that combines multiple articulatory targets and multiple emotion classes is proposed.

- Cross-corpus studies are conducted to evaluate the generalization property of this method across databases with different types of expressions and recording conditions.

1.4 Low-Memory Architectures for Keyword Detection

Keyword detection aims at identifying keywords of interest embedded in a continuous speech stream. By using keywords to initiate voice input, this feature enable users to have a fully hands-free interaction with their mobile devices. Such a voice trigger system needs to be in a continuously listening mode. This has serious implications on the battery life of mobile devices, hence, methods that facilitate keyword detection with minimal power consumption and resource usage are strongly desired for practical applications.

Keyword detection has been studied extensively in prior works [23, 24, 25, 26, 27, 28, 29, 30]. Recently, neural network based methods, inspired from deep learning, have shown tremendous success on speech recognition tasks compared to GMM-HMM systems [31, 32]. An extension of this approach for keyword detection was presented in [32]. The resulting network is quite large, requiring upto a few million multiplications every few milliseconds as well as large memory banks for storing these weights. Mobile devices are often constrained in the amount of available hardware resources, thus requiring further optimizations before being deployed on actual hardware.

1.4.1 Contributions

In this dissertation, a neural network-based architecture, followed by techniques to identify and eliminate the redundancy among the network weights, is proposed to address such constraints. The trade-off between detection accuracy and memory is assessed to evaluate performance.

The key contributions are as follows:

- A post-processing method to determine if a keyword is present in the phase, thus returning a global phrase-level prediction.
- An aggressive, fixed-point implementation scheme, which is shown to yield a comparable detection accuracy to a floating-point implementation, while, reducing the memory to as few as 200 KBs.

1.5 Ambient Sounds-based Lifelogging

Preservation and recollection of facts and events are central to human experience and culture, yet our individual capacity to recall, while astonishing, is also famously fallible [33]. As a result, technological memory aids date back to cave paintings and beyond. More recent trends include the shift from specific, active records (such as making notes) to transparent, comprehensive archives (such as the sent box of an email application) which become increasingly valuable as the tools for retrieving the contents improve [34].

To illustrate the concept of lifelogging and the challenges it may pose, consider the following scenario. A user, equipped with a wearable audio recorder or using his/her smartphone, is able to record audio for a single or multiple days. During this period, the user has a conversation with a friend, sees an interesting police car chase and hears some new piece of music on the radio. At some later point, the user wishes to share these events with a friend or simply wishes to recall what exactly happened. Manually browsing or searching for specific events through a long recording can be a time-consuming task. Hence, there is a strong requirement for computational techniques and frameworks that can perform automatic logging and the multiple subtasks [35]. Currently available techniques for various aspects of lifelogging, such as feature extraction, segmentation and annotation, are individually mature enough for real-world applications [36, 37, 35], yet their evaluation on long duration recordings, as found in lifelogging, is quite limited.

1.5.1 Contributions

In this dissertation, a complete framework for archival and retrieval of long duration recordings in a lifelogging scenario is presented. Experiments are conducted over data collected from a single subject to evaluate the proposed framework. The implementation aspects and complexities of a lifelogging application are modeled using a virtual platforms-based, top-down design methodology. Tools such as QEMU [38] and SystemC [39] are used to create such platforms. Optimizations are performed in an iterative fashion to address various design and system constraints such as power, speed, bandwidth, storage and accuracy.

The key contributions are as follows:

- Indexing and retrieval methods for long duration audio recordings are presented. Taking into consideration the nature of daily lifelogs, an augmented feature set is proposed to further improve the performance.
- A prototypical Android-based application, SoundBlogs, is presented to demonstrate important aspects of lifelogging.
- Using virtual platforms, a top-down design methodology is proposed. This allows hardware/software developers to gradually iterate from a high-abstraction, functional level towards a low-abstraction, refined hardware and software architecture, while, optimizing for system constraints at each iteration.
- A concept development kit (CDK), a packaging of the aforementioned tools, is

presented, thus, allowing designers to extend this methodology to their custom applications and develop prototypes rapidly.

The work related to this topic is reported in [40, 41].

1.6 Thesis Organization

The remainder of this dissertation is organized as follows: Latent topic model-based features for acoustic emotion recognition are described in Chapter 2. An extension of this method to multiple modalities is presented in Chapter 3. An articulation constrained learning approach is described in Chapter 4. Neural network optimizations for keyword detection on resource constrained hardware are presented in Chapter 5. A framework for indexing and retrieval of ambient sounds for lifelogging is described in Chapter 6, followed by virtual platform models in Chapter 7. Finally, summary and conclusions are outlined in Chapter 8.

Chapter 2

ACOUSTIC EMOTION RECOGNITION

Commercial applications of automatic emotion recognition include systems for customer services [42], call centers [43], intelligent automobile systems, and game and entertainment industries [44, 45]. Research disciplines, such as psychology, psychiatry, behavioral science, and neuroscience can benefit greatly from such systems. For example, they can help improve the quality of research by improving the reliability of measurements and speeding up the tedious and manual task of processing data [46, 47]. Other research areas that would reap substantial benefits from emotion recognition include studies related to social and emotional development research [48], mother-infant interaction [49], psychiatric disorders [47]. Automatic detection of emotional states and moods, including fatigue, depression, and anxiety, constitutes an important step toward personal wellness and assistive technologies [50].

At a high level, the problem of acoustic emotion recognition can be defined as the task of finding a mapping from input (speech) to output (emotions). Acoustic features extracted from speech can be examined at different resolution levels - frame, phoneme, syllable, word, sentence or even at the dialog level [51]. Similarly, various possibilities exist for representing emotions. Broadly, they can be described as (i) categorical, or, (ii) dimensional attributes [52]. In a categorical representation, emotions are described by discrete labels, such as happy, sad or angry. Whereas, in a dimensional representation, emotions are treated as points in a continuous, 2-D space comprising of arousal and valence. Here, arousal refers to the intensity level, valence refers to the pleasantness of the emotional state. Typically, the mapping is a multi-stage process that involves (i) extraction of emotionally relevant features, (ii) representation, and

(iii) classification/regression. The goal is to find a mapping which maximizes the performance according to pre-defined metrics, such as average recall or class-wise accuracies for classification, and, mean squared error or correlation coefficient in case of regression.

Although research on acoustic emotion recognition is progressing rapidly, most of the studies focus on only acted and prototypical emotions, demonstrating reasonable success [53]. Such studies are definitely important in order to identify and build generic templates of emotions, however, real-world applications require exhaustive evaluations under different criteria such as speaker independence, spontaneously expressed emotions and cultural or linguistic variations [54]. This task is challenging and difficult due to the varying degree and expression of emotions across speakers, moods, personalities, languages and cultures. Consequently, determining the appropriate set of features, representation and classification methods bear a strong impact on performance.

2.1 Background

2.1.1 Low-Level Descriptors

Low-level descriptors (LLD) or features that efciently characterize the emotional content of speech can be grouped into four types: prosodic, voice quality, spectral and custom. Prosodic features include pitch or fundamental frequency, energy-related features such as log energy or intensity and duration-related features such as zero-crossing rate and the voiced-to-unvoiced duration ratio [8, 55]. Voice quality features include jitter, shimmer which characterize the harshness, breath and tension related aspects of speech [56, 57]. Spectral features include formant-based features such as linear prediction coefficients (LPC), the energy of spectral sub-bands such as log frequency power coefficients (LFPC) [13], Mel fiterbanks (MFB) [10, 11] and Mel frequency cepstral coefficients (MFCC) [51]. Custom features include the Teager energy operator (TEO), which models the non-linear airflow patterns in the vocal system and its modulation under stressful conditions [58, 59]. In spite of the numerous choices, there is no clear winner; most approaches rely on a combination of different features. Prior research works [51, 60] have identified pitch, energy, duration, MFBs and MFCCs to be the most successful features.

2.1.2 Representation

The LLDs are extracted over the chosen unit of analysis (syllables, words or turns) using sliding and overlapping windows. This results in a continuous-valued, multi-dimensional trajectory of LLDs, say $[f_1, .., f_n, ..f_N]$. Here, $n = 1, .., N$ denotes the length of the unit, $f_n \in \mathcal{R}^p$ and p refers to the number of LLDs. Based on the method used to model the trajectory, representation methods can be broadly classified in two categories: segmental [13, 14] and supra-segmental [9, 10, 11, 12]. The former method attempts to directly model each frame f_n or the temporal information in the trajectory, i.e. transitions from f_n to f_{n+l}, where l is the order or range of the model. A higher l is usually required to capture the long range dependencies in emotional speech, which in turn, increases the training complexity.

Supra-segmental representations overcome this dependency limitation by finding global descriptors that characterize the long-term behavior [51]. Since this method aligns well with the way human beings perceive emotions, they have also shown to yield a better performance than segmental methods. Specifically, each turn is represented by a single, multi-dimensional vector $F \in \mathcal{R}^q$, where q refers to the dimension of the global descriptor and $q \gg p$. The global descriptors are found by extracting various statistics over the LLD trajectory. Common statistical functions include mo-

Table 2.1: A Typical Supra-Segmental Feature Set as Described in the 2009 Inter-Speech Emotion Recognition Challenge. 12 Functionals Applied over 16 LLDs and Their First-Order Derivatives Result in a Total of 384 Global Descriptors.

LLDs ($16 \cdot 2$)	Functionals (12)
Zero Crossing Rate (ZCR)	Mean
Harmonic Noise Ratio (HNR)	Standard Deviation
RMS Energy	Kurtosis, Skewness
Pitch (F0)	Extremes: value, position, range
MFCC 1-12	Linear regression: offset, slope, MSE

ments, extremes, percentiles, ranges, slope, linear regression coefficients, and many more. Emotion recognition challenges such as the 2009 InterSpeech Emotion Recognition Challenge [61] and the 2011/2012 Audio-Visual Emotion Recognition Challenge [62, 63] have provided an exhaustive set of global descriptors by performing a brute-force collection of such statistics. A typical feature set is shown in Table 2.1.

2.1.3 Classification

The methods used for classification or regression are dependent on the underlying representation. Classification over segmental representations is performed using generative models such as Gaussian mixture models (GMM) [64], hidden Markov models (HMM) [13] and their variants, e.g. Gaussian mixture vector autoregressive (GMVAR) models [14] and switching linear dynamical models [65]. A separate model is trained for each emotion category; class label is assigned based on the model for which the test instance achieves maximum likelihood (ML). Other dynamic approaches that are better at modeling long-term dependencies such as long short-term memory (LSTM) networks [66] and conditional random fields (CRF) [67] have also been used for segmental representations.

Discriminative classification/regression techniques have been found to combine well with supra-segmental representations. Typically, unsupervised or supervised feature selection is applied over the large feature sets to reduce their dimensionality. Few select methods include information gain [68], principal feature analysis [69], and deep belief networks [70]. Classification techniques include but are not limited to linear discriminant analysis or k-nearest neighbor [71], artificial neural networks (ANN) [72], decision trees [9], random forests [73] or support vector machines (SVM) [43, 12, 51, 64, 61]. In some cases boosting methods such as AdaBoosting have also shown to be successful [51, 74]. The ease of training combined with its reliable performance have made SVMs the ideal choice for classification. Continuous value prediction, i.e. regression is performed using linear or support vector regression [63].

Previous studies have shown that discriminative techniques combined with supra-segmental representations clearly outperform segmental approaches, making the former the de facto choice [12, 51, 62, 63, 64, 61]. Yet, the discriminative approach does not provide a clear, generative explanation of how emotions influence the perceived acoustic properties of speech. Brute-force collection of statistics works well as a data-driven approach, but it provides limited insight into the relationship between speech and emotions. On the other hand, segmental approaches are capable of providing generative explanations, yet, their inability to capture long-range dependencies leads to poor recognition performance.

2.2 Latent Topic Models

With an aim to address the aforementioned limitations, a novel method, based on latent topic models (LTM), is presented for the extraction of supra-segmental representations. Inspired by natural language processing, an alternative perspective towards emotion recognition is offered. Similar to contemporary supra-segmental methods, sequential or time-related information is ignored here, but, long term behavior is captured from the co-occurrence patterns among LLDs. This property is particularly useful for emotional speech as it often consists of multiple emotions expressed with varying degrees of strength and in an irregular temporal structure. More importantly, LTMs operate directly on LLDs and automatically learn features without requiring a brute-force collection.

In a typical LTM for text collections, each text document is assumed to comprise of a mixture of multiple topics [15]. And, each topic defines a discrete distribution over all the possible words in a dictionary. Given the observed words of a document, the latent topics are inferred from the co-occurring, repetitive patterns between words. Semantically similar documents are expected to exhibit a similar distribution over the latent topics, information that is used for categorizing or classifying documents. Apart from their obvious application in natural language processing, LTMs have been used for human activity recognition [75], image annotation and segmentation [76, 77], image-based object recognition [78] and acoustic scene analysis [79].

Latent semantic analysis (LSA) [80] and its stochastic counterpart, probabilistic LSA (pLSA) [81], are two of the earliest topic models. Latent Dirichlet allocation (LDA) [15] is an extension of pLSA to a Bayesian framework by placing a Dirichlet prior over the topics. Although LDA is quite useful for unsupervised topic discovery and qualitative interpretation, its performance in classification-related tasks is limited. Recently, undirected graphical models, based on the idea of distributed representations, have shown to perform better than LDA. These models are variants of the energy-based restricted Boltzmann machines [82, 83], such as replicated softmax model [16] or the constrained Poisson model [84]. Due to their popularity and performance, LDA and RSM are considered in this study.

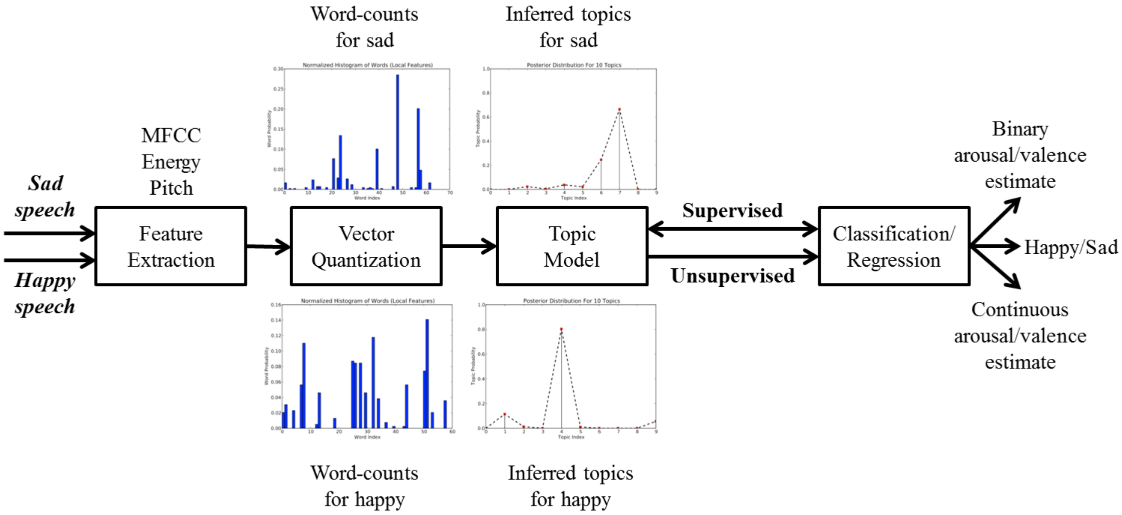


Figure 2.1: Block Diagram of the Proposed Feature Extraction and Classification Framework Using LTMs for Acoustic Emotion Recognition.

2.2.1 Notation

A document d is a sequence of N words such that $d = (v_1, ..., v_N)$. A corpus is a collection of D such documents, $d_1, ..., d_D$. Each word v_n is defined to be an item from a dictionary indexed by $\{1, ..., K\}$. Words are represented as unit vectors, where $v_{nk} = 1$ if the n^{th} word belongs to the k^{th} dictionary element. h is the J-dimensional latent topic vector inferred from the observed words in a document. The distribution of words over topics is denoted by the $J \times K$ parameter matrix W.

2.2.2 Overview of Proposed Approach

A block diagram overview of the proposed framework is shown in Figure 2.1. The LLDs are extracted over the entire duration of the turn using the following procedure: Raw speech is high-pass filtered with a pre-emphasis coefficient of 0.97. Hamming windows of duration 25 ms are used to extract features at a rate of one feature vector every 10 ms. LLDs such as energy, fundamental frequency (F0), and the first 12 MFCCs (ignoring the 0^{th} coefficient) are extracted. The first and second-order differences are appended to obtain a 42-D feature vector per frame. Energy and MFCCs are extracted using the HTK Toolkit [85], while the F0 estimates are extracted using the OpenEar Affect Recognition Toolkit [12]. Principal component analysis (PCA) is further applied to reduce the dimensionality to 13 features.

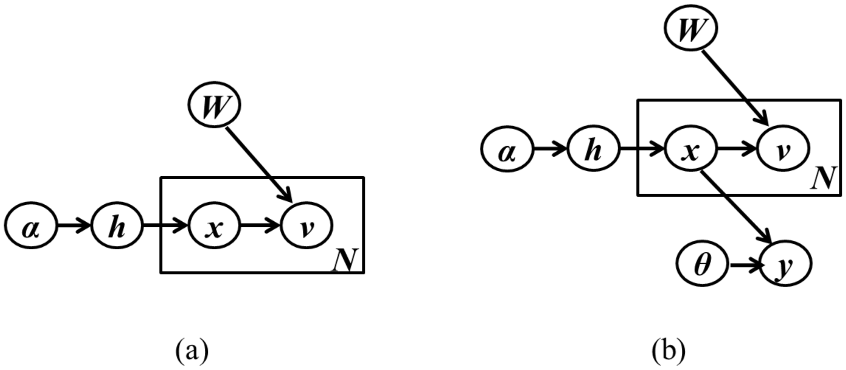
Each turn is represented as a multi-dimensional, continuous-valued trajectory of low-level acoustic descriptors, $f \in \mathcal{R}^{N \times p}$. Here N is the number of frames and p refers to the dimensionality of the feature vector. Topic models require these features to be converted to discrete values or symbols, analogous to words in a text document. A dictionary of K candidate feature vectors, $\{f_1^*, ..., f_K^*\}$, is constructed using the LBG-VQ algorithm. Each frame is then mapped to the the dictionary element f_k^* it is closest to in terms of the Euclidean distance and denoted by the corresponding index k. As a result, each turn or document d in a collection of D documents is represented as a stream of words $v_1, ..., v_N$ corresponding to the feature trajectory $f_1, ..., f_N$. Each word, v_n, is a size K vector, where $v_{nk} = 1$, if the n^{th} word belongs to the k^{th} dictionary element. The choice of the dictionary size is $K \in \{64, 128, 256, 512\}$.

During the recognition phase, latent topics are inferred from the observed words in a turn. Using these topics as features, the classifier produces a binary, categorical or continuous response depending on the application.

2.2.3 Latent Dirichlet Allocation

LDA [15] is a generative probabilistic model for a corpus. A graphical model for the same is shown in Figure 2.2 (a). The generative process for each document d in a corpus is as follows.

- Choose $h \sim Dirichlet(\alpha)$



(a)

(b)

Figure 2.2: Graphical Model Representation of (a) LDA, and (B) sLDA. Plates (Rectangular) Drawn Around Nodes Indicate Replication, Which Corresponds to the Number of Input Observations or Words in a Document i.e. N.

- For each word v_n in the document -
  - Choose a topic $x_n \sim Multinomial(h)$
  - Choose a word $v_n \sim p(v_n|x_n, W)$

Here, x_n is a J-dimensional unit-basis vector indicating the topic active for the currently observed word v_n. α is a corpus-level hyperparameter sampled once for the entire collection of documents. Parameters α, W, the dictionary size K and number of topics J are estimated and fixed during training. During inference, the latent variables h and x are estimated given the observed words for each document as given in Eq (2.1).

$$p(h, x|v, \alpha, W) = \frac{p(h, x, v, \alpha, W)}{p(v|\alpha, W)} \tag{2.1}$$

Exact inference in LDA being intractable as it involves marginalization over latent variables, different techniques based on Markov Chain Monte Carlo (MCMC) sampling or variational approximation have been proposed in literature [86]. The latter approach is used here as it allows for faster inference [15]. Briefly, the posterior is

```
Initialize $\phi_{nj} = 1/K$ for all $j$ and $n$.
Initialize $\gamma_j = \alpha + N/J$ for all $j$.
repeat
    for $n = 1 : N$ do
        for $j = 1 : J$ do
            $\phi_{nj} = W_{jv_n} exp(\Psi(\gamma_j))$
        end for
        Normalize $\phi_n$.
    end for
    $\gamma = \alpha + \sum_{n=1}^{N} \phi_n$
until convergence
```

Figure 2.3: A Variational Approximation Algorithm for Inference in LDA.

modeled by a variational distribution $q(h, x|\gamma, \phi)$. γ and ϕ are free variational parameters, iteratively used to minimize the Kullback-Leibler (KL) divergence between q and the posterior, as in Eq (2.2).

$$(\gamma^\star, \phi^\star) = \arg\min_{(\gamma,\phi)} D(q(h, x|\gamma, \phi)||p(h, x|v, \alpha, W)) \tag{2.2}$$

An outline of the algorithm to infer (γ, ϕ) and consequently (h,x) is described in Figure 2.3. Here, Ψ denotes the digamma function obtained by taking the first-derivative of a log-gamma function, i.e. $\frac{d\ln(\Gamma(\gamma))}{d\gamma}$. Learning of parameters α and W from training examples is performed using the Expectation-Maximization (EM) algorithm.

In order to perform binary or multi-class, categorical emotion recognition, a softmax regression-based classifier is trained over the posterior topics h inferred from the training examples. The classifier parameters, θ, are estimated by minimizing the cross-entropy error with standard L2 regularization, as per Eq (2.3) -

$$L(\theta) = -\frac{1}{S}\left[\sum_{s=1}^{S}\sum_{c=1}^{C} 1\{t^{(s)} = c\} \log p(y^{(s)} = c|h^{(s)}, \theta)\right] + \frac{\lambda}{2}\|\theta\|_2^2 \tag{2.3}$$

Here, $1\{\cdot\}$ is the indicator function, S denotes the number of training examples, C denotes the number of classes, $t^{(s)}$ denotes the ground truth for example s, and λ denotes the regularization parameter. Iterative minimization is performed using mini-batch stochastic gradient descent with a batchsize of 100, and a learning rate and momentum of 0.005 and 0.8, respectively. The output, $y^{(s)}$, defined by the softmax function in Eq (2.4) -

$$p(y^{(s)} = c|h^{(s)}, \theta) = \frac{\exp(\theta_c^T h^{(s)})}{\sum_{l=1}^{C}\exp(\theta_l^T h^{(s)})} \tag{2.4}$$

This returns the posterior probability for each class. The label is then predicted by evaluating $\operatorname{argmax} p(y^{(s)} = c|h^{(s)}, \theta)$. Similar expressions can be derived for the case of predicting real-valued outputs using linear regression.

2.2.4 Replicated Softmax Models

RSM belongs to the family of undirected, energy-based models known as restricted Boltzmann machines (RBM) [82]. The visible unit is modeled as a softmax variable instead of a Bernoulli variable as in RBM [16]. A graphical representation of this model is shown in Figure 2.4 (a). For a turn with N words, the observation v forms an $N \times K$ binary matrix, and $h_j \in \{0, 1\}$ are the binary stochastic latent topics. The energy of this configuration is defined as in Eq (2.5).

$$E(v, h) = -\sum_{n=1}^{N}\sum_{j=1}^{J}\sum_{k=1}^{K} W_{njk} h_j v_{nk} - \sum_{n=1}^{N}\sum_{k=1}^{K} v_{nk} a_{nk} - \sum_{j=1}^{J} h_j b_j \tag{2.5}$$

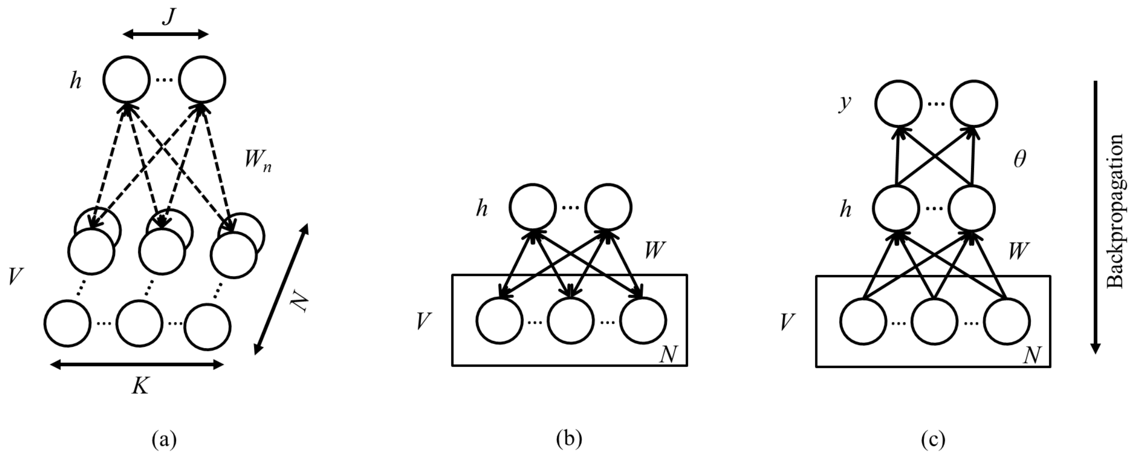


Figure 2.4: Graphical Model Representation of (a) RSM Without Weight-sharing, (b) RSM after Weight-sharing, and (c) sRSM. In (a), the Weights Are Only Shown for the nth Word. Plates (Rectangular) Drawn Around Nodes in (b) and (b) Indicate Replication, Which Corresponds to the Number of Input Observations or Words in a Document, i.e. N.

W_{njk} is a symmetric interaction term between visible unit n that takes on value k, and hidden topic j; b_j is the bias of hidden topic j and a_{nk} is the bias of visible unit n that takes on value k. In addition to the energy of the joint configuration, a RSM is fully defined by the conditional probabilities of the visible and hidden units with respect to each other, i.e. Eqs (2.6) and (2.7). Here, $\sigma(x) = 1/(1+\exp(-x))$ is the logistic function.

$$p(v_{nk} = 1|h) = \frac{\exp(a_{nk} + \sum_{j=1}^{J} h_j W_{njk})}{\sum_{q=1}^{K} \exp(a_{nq} + \sum_{j=1}^{J} h_j W_{njq})} \tag{2.6}$$

$$p(h_j = 1|v) = \sigma(b_j + \sum_{n=1}^{N} \sum_{k=1}^{K} v_{nk} W_{njk}) \tag{2.7}$$

Ignoring the sequence in which words arrive, if the k^{th} unit for each word v_n is forced to share its weight with the k^{th} unit of all the other words in the turn, then W_{njk} can be written simply as W_{jk}. This allows the model to account for turns of different durations, which is crucial for turn-based practices. The energy in Eq (2.5) can now be rewritten as in Eq (2.8) -

$$E(v,h) = -\sum_{j=1}^{J}\sum_{k=1}^{K} W_{jk}h_j\hat{v}_k - \sum_{k=1}^{K}\hat{v}_k a_k - N\sum_{j=1}^{J} h_j b_j \tag{2.8}$$

Here, $\hat{v}_k = \sum_{n=1}^{N} v_{nk}$ denotes the frequency with which the k^{th} dictionary element appears in the turn. Exact maximum likelihood-based learning of parameters W, a and b is intractable in this case, hence, an approximate technique called *contrastive divergence* (CD) algorithm is used. Further details on this technique and its convergence properties can be found in [82]. Using CD, the update equation for the weights W is given in Eq (2.9). Similar update equations can be derived for the bias terms.

$$\Delta W_{jk} = \eta(E_{data}[\hat{v}_k h_j] - E_{model_T}[\hat{v}_k h_j]) \tag{2.9}$$

Here, η is the learning rate and E_{model_T} represents the expectation with respect to the distribution after running a Gibbs chain for T steps. $T = \infty$ is equivalent to maximum likelihood learning. Usually, a small value of T (1, in this case) is adequate for generating abstract features [16].

So far, the LTM-based features are derived in a completely unsupervised manner. Consequently, the inferred topics or features are not naturally suited for discriminative tasks. Supervised learning via a supervised LDA (sLDA) model and a supervised RSM (sRSM) is presented below to further improve the performance.

2.2.5 Supervised Latent Dirichlet Allocation

Supervised LDA (sLDA) [77] differs from its unsupervised counterpart in the following aspect: an additional node y, the class label or output, is introduced as shown in Figure 2.2 (b). The output is predicted according to Eq (2.10) -

$$y_m \sim \frac{\exp(\theta_m^T \bar{x})}{\sum_l^C \exp(\theta_l^T \bar{x})} \tag{2.10}$$

Here, $\bar{x} = \sum_{n=1}^{N} x_n$ represents the empirical topic frequencies. The variable θ, which parametrizes the relationship between the topic indicators and the output label, is estimated in an iterative manner along with the topics. As a result, class-specific information is considered while learning topics, thereby leading to better discrimination in comparison to LDA. Variational approximation is used to infer the latent variables as described in [77]. The class label is predicted by evaluating $\operatorname{argmax} \theta_m^T \bar{x}$. Eq (2.10) is specific to binary or categorical classification; a similar expression can be derived for regression.

2.2.6 Supervised Replicated Softmax Models

In order to extend an RSM for supervised learning, an sRSM is proposed. This is essentially a feed-forward neural network with its initial weights obtained from an unsupervised RSM. The weights are then fine-tuned for discrimination using backpropagation. As opposed to random initialization, the RSM is treated as a pre-training stage, which learns topics that initially capture properties of the underlying input only. Backpropagation then serves to slightly perturb and refine these topics with respect to the output labels. This process facilitates the extraction of topic-features that are optimal for discriminative tasks. RBMs for learning discriminative features have been previously described in [87, 88], where, a deep neural network-based generalized discriminant analysis (DNN-GerDA) was used to learn emotion-specific, turn-level features. The proposed sRSM is fundamentally different in the following key aspects: (i) DNN-GerDA employs the Fisher discriminant criterion, which maximizes the ratio of between-class variance to within-class variance, while sRSM directly minimizes the cross-entropy error, which is more appropriate for classification-related tasks [89, 90],

(ii) DNN-GerDA assumes that the extracted features are drawn from Gaussian class-conditional distributions, while sRSM makes no assumptions regarding the statistical properties of the inferred topics, (iii) DNN-GerDA accepts arbitrarily distributed, real-valued observations as input, whereas, sRSM models discrete, count-like observations commonly found in text collections, and (iv) the discriminative features in [88] are learnt over turn-level statistics extracted via brute-force as opposed to the acoustic bag-of-words used in this work.

An sRSM is depicted as a graphical model in Figure 2.4 (c). The input and hidden layer are the same as that of an RSM, while the topmost layer performs output prediction. For C-class, categorical recognition, the top layer is a softmax layer and the output is computed via Eq (2.11) -

$$p(y_m|h) = \frac{\exp(\theta_m^T h)}{\sum_{l=1}^{C} \exp(\theta_l^T h)} \tag{2.11}$$

For backpropagation, the cross entropy error is used as the cost function for classification, and the mean squared error (MSE) for linear regression. Stochastic gradient descent is used to update the parameters with a learning rate and momentum of 0.005 and 0.8, respectively.

The advantage of using pre-trained weights as opposed to random initialization is shown in Figure 2.5, which shows the classification error across epochs, averaged over 100 runs. The results are displayed for the task of arousal-based, binary classification on two databases, SEMAINE and USC IEMOCAP. It is clearly evident that back-propagation over randomly initialized weights is prone to get stuck at a bad local optima and yield a higher classification error. On the contrary, pre-training using an RSM, first models the observations in an unsupervised manner and finds a good starting point for the weights, which leads to a lower classification error.

It is important to note that the impact of pre-training is usually higher for smaller

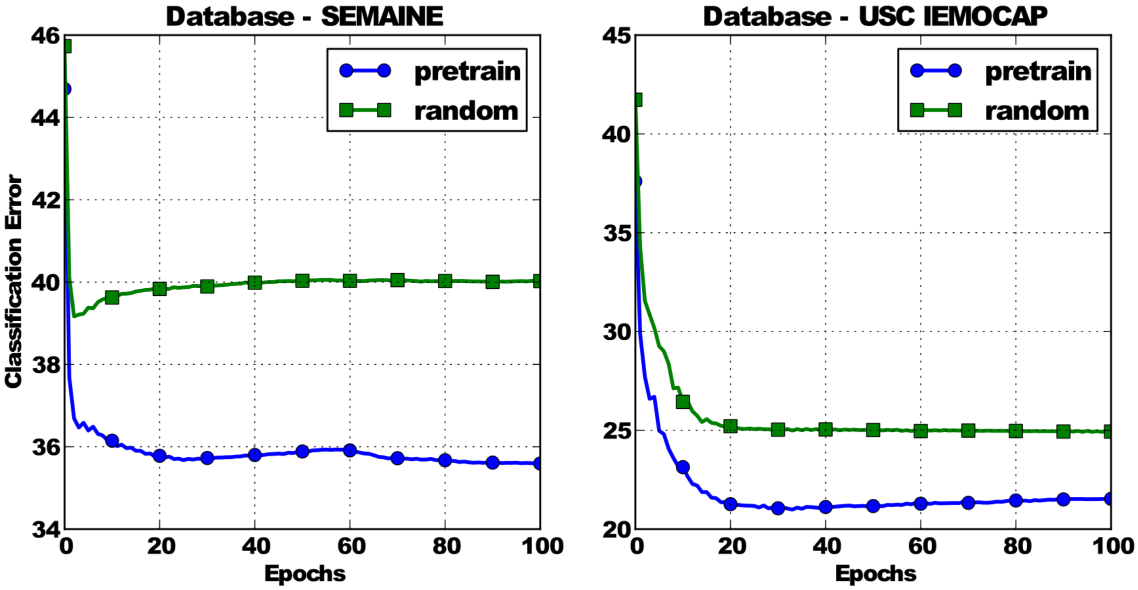


Figure 2.5: A Comparison of the Classification Error (%) Between an sRSM with Pre-trained and Randomly Initialized Weights on SEMAINE and USC IEMOCAP. The Higher Error Achieved in the Latter Case Is Due to the Parameters Getting Stuck at a Bad Local Optima. Pre-training Overcomes This Limitation and Provides a Better Starting Point Using the Input Observations Only.

databases. Here, the SEMAINE and USC IEMOCAP databases consist of approximately 1000 and 5000 training examples, respectively. From Figure 2.5, one can observe that there is a slight decrease in the effectiveness of pre-training from SEMAINE to USC IEMOCAP. To further investigate this behavior, the number of training examples used for fine-tuning the sRSM was gradually increased. Using the same database, USC IEMOCAP, Figure 2.6 shows that the difference in classification error between pre-trained and randomly initialized sRSM gradually declines as the training examples increase from 500 to 5000.

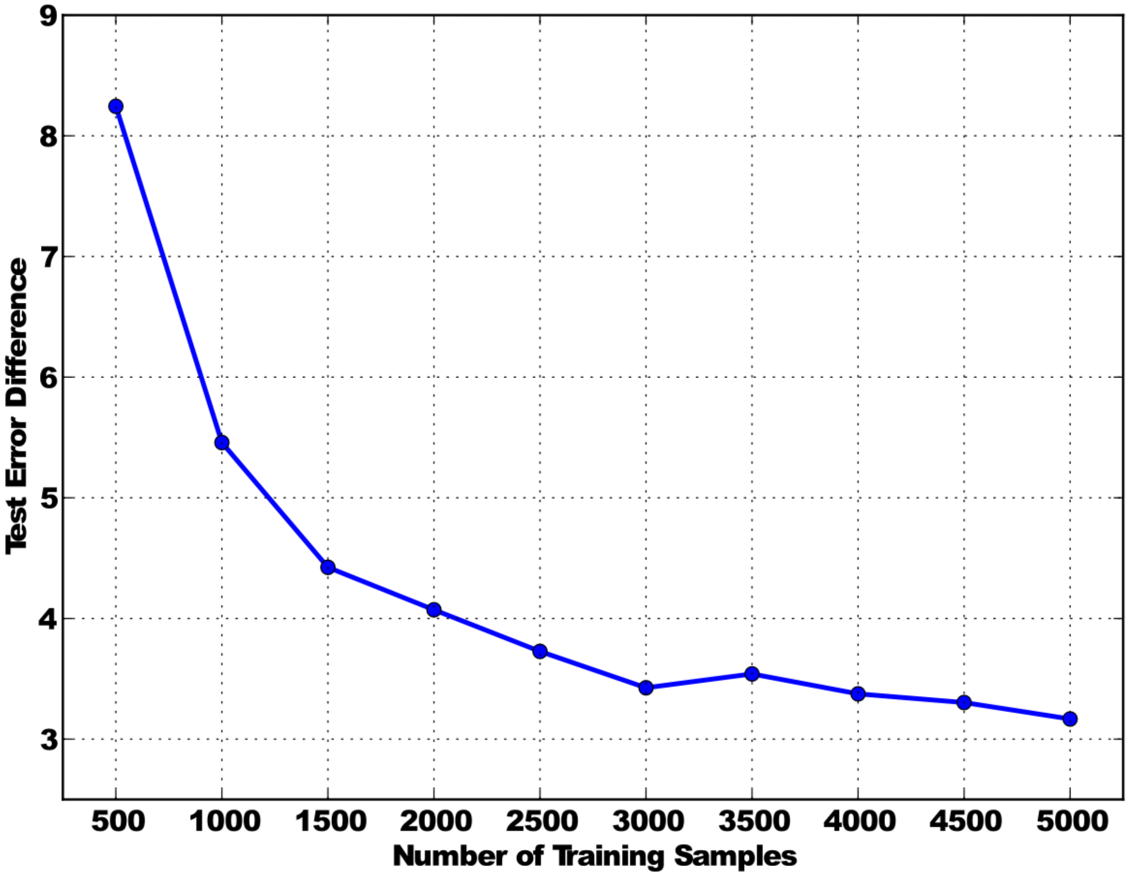
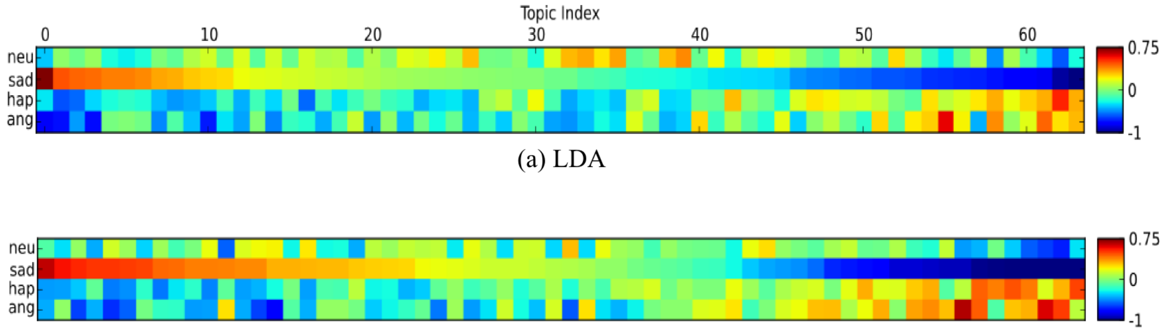


Figure 2.6: Impact of Pre-training Using Training Sets of Different Sizes. The Difference in Classification Error (%) Between an sRSM with Pre-trained and Randomly Initialized Weights on USC IEMOCAP. Note the Gradual Decline in the Difference as the Number of Examples Available for Fine-tuning the sRSM Is Increased.

2.3 Qualitative Interpretation

It is worthwhile to investigate and provide a physical interpretation of topics in terms of their relationship with emotions and the underlying acoustic words. Ideally, one would expect to have as many topics as emotion categories, but, variations across speakers, spoken content and mannerisms cause the number of topics, J, to usually lie between the number of emotions, C, and the dictionary size, K. Among these topics, only a few may convey emotion-specific information, while certain topics may



(a) LDA

(b) RSM

Figure 2.7: Qualitative Interpretation of Emotions and Topics Using (a) LDA, and (b) RSM. The Normalized PMI Between 64 Topics and 4 Emotions Reveals That Individual Topics Capture Emotion-Specific Information.

be present across all emotions and can be considered uninformative or irrelevant. In order to identify such topics, a normalized point-wise mutual information (PMI) measure between the individual topics h and emotions y is proposed. This measure quantifies the discrepancy between the joint probability of h and y and their marginal distributions under the assumption of independence. The PMI is computed as per Eq (2.12), where $1 \leq j \leq J$ and $1 \leq c \leq C$.

$$pmi(h_j; y_c) = log \frac{p(h_j, y_c)}{p(h_j)p(y_c)} \tag{2.12}$$

The values are further normalized to a range of $[-1, +1]$ as described in [91] using Eq (2.13).

$$npmi(h_j; y_c) = \frac{pmi(h_j; y_c)}{-logp(h_j, y_c)} \tag{2.13}$$

Here, -1 indicates never co-occurring, $+1$ indicates always occurring together, and 0 indicates independence. The most informative topics for each emotion are then identified and ranked by the decreasing order of their normalized PMI values.

In Figure 2.7, the normalized PMI values for 64 topics, extracted using unsupervised LDA and RSM, for a single male speaker across the four emotions (neutral, sad, happy, angry) of USC IEMOCAP are displayed. Topics that exhibit a high co-occurrence with sadness are found to occur never or rarely with angry or happy emotions. Similar observations can be made for the vice versa case. Hence, even without using any label information while learning topics, one can observe that the emotions are nicely separated in the topic space, with sad and happy topics being the most easily distinguishable from each other. The only exception is neutral; very few topics show a high co-occurrence with only neutral emotions, with a majority of them also co-occurring with the other three emotions. Between LDA and RSM, one can observe that the topics obtained via LDA capture neutral emotions slightly better than RSM, i.e. the highly ranked topics for neutral emotions co-occur less with other emotions. On the other hand, RSM represents the remaining emotions such as sad, happy and angry better than LDA.

Taking further advantage of the generative mechanism of topic models, an interpretation of the relationship between topics and the underlying acoustic words can also be provided. Using the weight matrix W characterizing $p(v|h)$, the most probable words under the highest ranked topic for each emotion are extracted. The spectrograms, reconstructed from the MFCCs, for the top 3 words are shown in Figure 2.8. For acoustic words grouped under sad or neutral topics, most of the energy is concentrated at lower frequencies (<2000Hz). In comparison, words grouped under happy or angry topics show that the energy is more spread out across frequency. Thus, it can be said that the individual topics induce a natural grouping over acoustic words primarily based on their energy distribution across different frequencies.

Evident from the normalized PMI values and the most likely words for individual topics, there is a strong overlap between happy-angry and neutral-sad emotions. The inability to visualize distinguishable characteristics across valence is a well-known

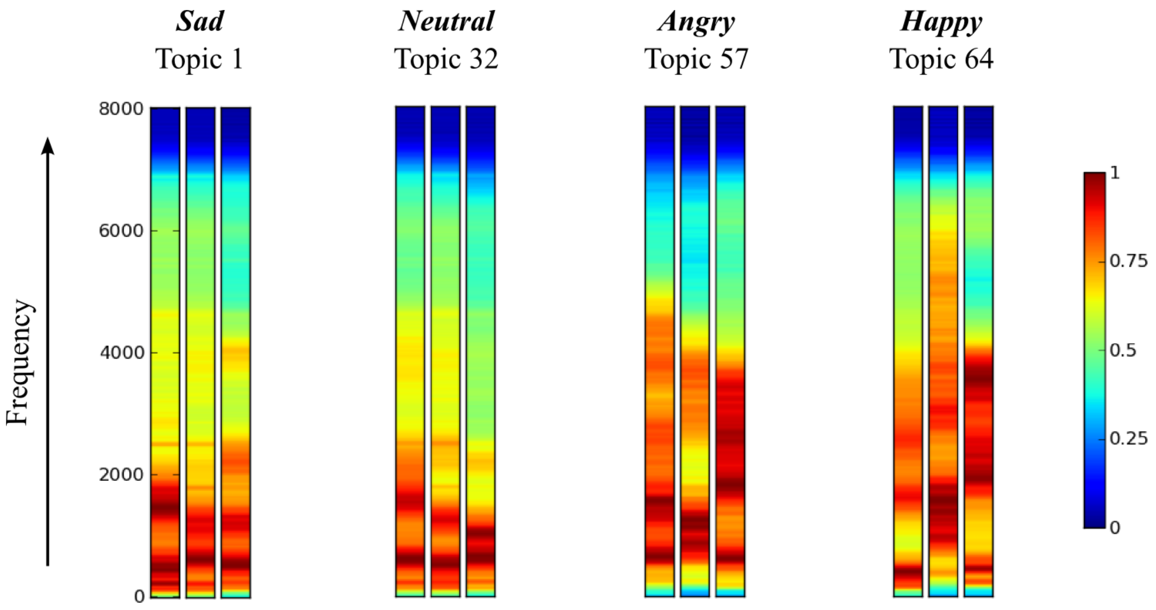


Figure 2.8: The Top 3 Probable Acoustic Words for the Highest Ranked Emotion-specific Topics Obtained Using RSM. Individual Topics Induce a Natural Grouping over Acoustic Words Based on Their Underlying Distribution of Energy Across Frequency.

limitation of speech, which is more reactive to changes along the arousal dimension. Experiments described in the subsequent sections will highlight the importance of LTMs, which allow us to represent each turn as a mixture of multiple emotion-specific topics, towards classification, especially along the valence dimension.

2.4 Databases

An important issue to be considered in the evaluation of an emotional speech recognizer is the degree of naturalness of the database used to assess its performance. Databases available for emotion recognition studies can be broadly classified in two categories based on the manner in which emotions are elicited from the speakers. The first category includes acted and prototypical emotions performed by professional actors or non-expert human beings. Select examples of such databases include EMO-DB

Table 2.2: Distribution of Emotions in EMO-DB.

Emotion	neu	ang	hap	fea	sad	bor	dis	Total	Speakers
# Turns	78	127	64	55	53	38	79	494	10

[92], DES [93] and eNTERFACE [60]. The goal of such databases is mainly to obtain generic definitions of emotions and to study their modulation effects on speech. The second category includes artificially induced emotions through human-machine or human-human interaction or spontaneously expressed emotions as in typical conversations. Such emotions exhibit a high degree of naturalness (realistic) but are hard to characterize and classify. Select examples of such databases include VAM [94], SEMAINE [95] and FAU AIBO [96]. USC IEMOCAP [97] is a unique database which covers both types of emotions. Although most of the databases are for private or commercial use, there is a growing number of freely and publicly available databases. Taking into consideration the types of emotions along with the number of speakers, size and quality of annotations, EMO-DB, USC IEMOCAP and SEMAINE are chosen for our experiments. Details of each database are provided below.

2.4.1 EMO-DB

The German language-based emotional speech database (EMO-DB) [92] consists of non-spontaneous, acted emotions by 10 speakers, 5 male and 5 female. Each utterance is labeled by a single emotion belonging to one of seven categories - *neutral (N)*, *happiness (H)*, *anger (A)*, *fear (F)*, *sadness (S)*, *boredom (B)* or *disgust (D)*. Only 494 utterances with a minimum of 80% human recognition accuracy and 60% naturalness are selected for our experiments. They are distributed across 7 emotions as shown in Table 2.2.

Table 2.3: Distribution of Emotions in USC IEMOCAP.

Emotion	neu	sad	hap	ang	Total	Speakers
# Turns	1708	1084	1636	1103	5531	10

2.4.2 USC IEMOCAP

The USC IEMOCAP [97] corpus was created by selecting 5 pairs of male-female actors to elicit emotions either by reading from a script or via improvisation in a conversational setting. There are a total of 151 dialogs which, after turn-based segmentation, yield a total of 10039 turns (5255 scripted, 4784 improvised). At least three evaluators assigned a categorical and a dimensional attribute to each turn. Categorical emotions include *neutral*, *sad*, *happy*, *excited*, *angry*, *frustrated*, *surprised*, *disgusted*, *afraid* or *xxx* (unknown). Dimensional attributes include arousal, valence and dominance-based continuous values.

Only the prototypical turns of USC IEMOCAP are selected for our experiments. Here, prototypical turns are those for which a majority consensus was obtained among evaluators. The goal is to evaluate the performance of a multi-class recognition system with a categorical output. Following the selection criteria outlined in [11, 9], turns labeled as *neutral*, *sad*, *happy*, *excited* and *angry* are only selected, while *happy* and *excited* are treated as the same emotion and merged as one class [9]. This results in a total of 5531 turns distributed across 4 emotions as shown in Table 2.3.

2.4.3 SEMAINE

The SEMAINE [95] corpus, intended for the study of interaction between humans and artificial agents, is recorded by engaging speakers in conversations with human operators. The latter are supposed to role-play characters with specific emotional traits. The four characters are Poppy (Happy), Prudence (Neutral), Spike (Angry)

Table 2.4: Distribution of Emotions in SEMAINE.

Task	Train Set	Dev. Set	Test Set
Arousal	1185	960	673
Valence	1129	936	673
# Speakers	8	7	6

and Obadiah (Sad). The emotions in this case are elicited spontaneously from speakers, who react to the operator's behavior. There are 24 speakers in the set, and each speaker interacts with almost all four characters, resulting in a total of 95 sessions. Of these, only 82 sessions include a force aligned transcript necessary for turn-based segmentation. Our experiments are designed according to the Audio-Visual Emotion Challenges held in 2011 [62] and 2012 [63], except for a few differences. First, emotions in these challenges were studied at the word-level contrary to the turn-level granularity adopted in this work. Second, only arousal and valence attributes are used here, while ignoring dominance and expectation. Each interaction is annotated by 2 to 8 evaluators at the frame-level (every 20ms). For each attribute - first, the average value across all evaluators is calculated for each frame. This is followed by an average of the values over all the frames in a turn to yield a single value. For continuous regression, these values are considered as ground truth, while for binary classification, the mean and thresholding process for USC IEMOCAP is followed here. Closely resembling AVEC 2011 and 2012, the 82 sessions and the corresponding 2818 turns are partitioned into 3 speaker-disjoint sets: train (1185), development (960) and test (673). Due to the unavailability of valence attributes for some turns, the training partition for valence-related experiments consists of 1129 turns.

2.5 Baseline and Metrics

In addition to comparison between different LTMs (LDA, sLDA, RSM and sRSM), two baseline feature sets are considered - IS09 and VQ. The former comprises of a set of 384 brute-force based statistical features and is the popular choice among researchers as evident from its use in the 2009 InterSpeech and 2011/2012 AVEC challenges [61, 62, 63]. These features were extracted using the openEAR Affect Recognition toolkit [12]. Classification is performed via a linear kernel SVM/SVR trained using the WEKA toolkit [98]. Results for the second baseline approach, VQ, are obtained via a linear kernel SVM/SVR trained directly over the acoustic word occurrences, i.e. bag-of-words representations.

The weighted average (WA) and unweighted average (UA) recall, defined in [61], are used as metrics to evaluate binary or categorical recognition performance. The former is defined as the average classification accuracy, while the latter is defined as the average of the class-wise accuracies. These metrics are calculated using Eq (2.14).

$$WA = 100 \cdot \frac{m}{M} \qquad UA = 100 \cdot \frac{1}{C}\sum_{c=1}^{C}\frac{m_c}{M_c} \tag{2.14}$$

Here, m denotes the number of examples correctly classified, and M denotes the total number of examples. Similarly, m_c and M_c denote the number of examples correctly classified and total number of examples, respectively, for a specific class c. Since UA recall is more appropriate for unbalanced datasets, statistical significance over the baseline is determined using a one-tailed test (difference of proportions) over the UA recall values. Unless mentioned otherwise, the significance level is at $\alpha = 0.05$. The correlation coefficient (COR) is used to evaluate performance for regression tasks.

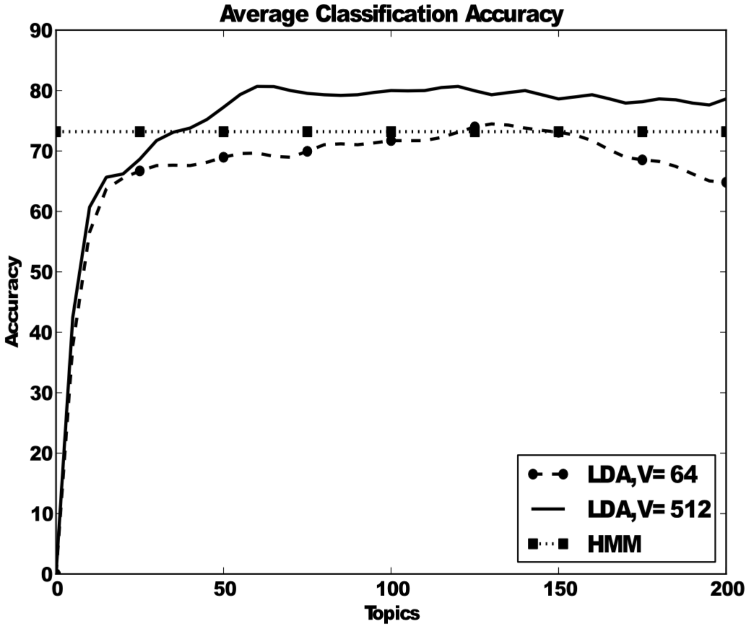


Figure 2.9: WA Recall for Emotion Recognition on EMO-DB and Its Variation with the Number of Topics and Dictionary Size.

2.6 Within-Corpus Evaluations

2.6.1 EMO-DB

The 494 turns of EMO-DB are split into two partitions, train and test, with speaker overlap between the two sets. 70% of the turns are used for training, while the remaining 30% are used for testing or evaluation. As shown in Figure 2.9, for a small-sized vocabulary, $K = 64$, LDA performs worse than HMM for most values of J, while approaching the baseline accuracy in certain cases. For a larger dictionary, $K = 512$, LDA always performs better than HMM once J exceeds 50 topics. For J less than 50, the topics fail to efficiently capture the statistical information between features. As the dictionary size K increases, VQ distortion reduces, thus, leading to improved partitioning and a better dictionary. This is evident from the improvement in recognition performance as K increases from 64 to 512. Specifically, for $K = 64$, a maximum classification accuracy of 74.1% is achieved with $J = 130$. While for $K = 512$, the accuracy is significantly higher at 80.7%. In this case, the number of topics is also fewer, $J = 60$. A relative improvement of 10.54% is obtained over HMM-based recognition methods. By ignoring the temporal structure, LDA is able to model higher-order dependencies in data compared to HMMs and yield a better recognition accuracy.

Classification performed directly over words obtained after VQ yields a WA recall of 65.2% for a dictionary of size $K = 512$. Recall purely based on chance is 25.70%. This further demonstrates the feasibility of a simple bag-of-words representation as well as the importance of extracting intermediate-level features via LTMs for emotion recognition.

Table 2.6.1 describes the confusion matrix between different emotions using LDA. If emotions are grouped into two distinct categories based on high (A, H, F) and low (N, S, B, D) arousal, it can be observed that misclassification is more prominent within a group than across groups. For example, 30% of happiness-related utterances are classified as anger and fear, whereas only 10% of the utterances are classified as low-arousal emotions such as neutral or disgust. These results are consistent with findings in previous works [13].

2.6.2 USC IEMOCAP

To ensure speaker independence, experiments were performed using a Leave-One-Speaker-Out (LOSO) strategy, resulting in a 10-fold process corresponding to the 10 speakers. For each fold, the LLDs are normalized such that the test partition has the same mean as that of the training partition. Dictionaries of multiple sizes, $K \in \{64, 128, 256, 512\}$, are learnt, while the number of topics $J \in [K/4, K/2]$. The

Table 2.5: Normalized Confusion Matrix for LDA on EMO-DB.

	N	A	H	F	S	B	D
N	**0.64**	0	0	0	0.13	0.23	0
A	0	**1.00**	0	0	0	0	0
H	0.05	0.25	**0.60**	0.05	0	0	0.05
F	0.12	0.12	0	**0.70**	0	0.06	0
S	0.06	0	0	0	**0.94**	0	0
B	0.18	0	0	0	0.05	**0.77**	0
D	0	0	0	0	0	0	**1.00**

Table 2.6: Recall Values (%) for Categorical Recognition on USC IEMOCAP. Symbols $*$ and $\dagger$ Indicate Statistical Significance over IS09 and VQ, Respectively.

Metric	IS09	VQ	LDA	sLDA	RSM	sRSM
neu	53.62	39.64	51.70	52.46	49.88	56.38
sad	62.45	71.31	66.79	67.34	74.72	70.39
hap	47.00	54.10	50.37	51.53	54.77	54.76
ang	57.57	49.95	52.04	52.95	52.31	54.58
WA	54.17	52.18	54.33	55.20	56.68	**58.29**
UA	55.14	51.94	55.22†	56.07†	57.92*†	**59.03***†

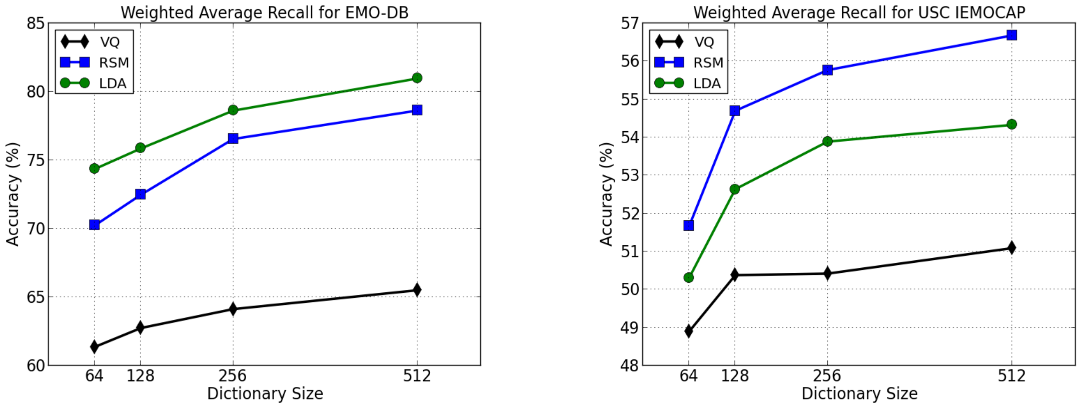
optimal values of K and J are highly dependent on the training examples and vary for each fold. Cross-validation over the training set was used to determine their optimal values.

The WA and UA recall values are presented in Table 2.6. Classification over acoustic words, i.e. VQ, yields a reasonable performance of 52.18%. LDA and RSM show a relative increase of 6.31% and 11.51%, respectively, over VQ. Supervised learning leads to further improvements as demonstrated by the better recall obtained using sLDA and sRSM over their respective unsupervised counterparts.

Compared to IS09, LDA shows a marginal improvement, however, RSM and sRSM demonstrate significant improvements with a recall of 57.92% and 59.03%, respectively. Relatively, RSM and sRSM offer gains of 5% and 7% over IS09, respectively. Compared to previous works, the proposed method clearly outperforms the UA recall of 50.69% achieved using HMMs in Metallinou et al. [11]. It also compares well with the previous best result of Lee et al. [9], where a recall of 58.46% was achieved using the IS09 feature set combined with a hierarchical decision-tree based classifier. Since the train and test partitions in these works differ slightly from the experiments described in this work, an exact comparison is not feasible.

Further inferences can be drawn from the class-wise accuracies provided in Table 2.6. It can be seen that speech-based methods are best at recognizing sadness. In case of neutral emotions, sRSM shows a 56.38% accuracy, which is better than the previous best of 54.54% obtained by Lee et. al [9] and 35.23% of Metallinou et. al [11]. The difficulty in recognizing neutral emotions is also evident from the inter-evaluator agreement - of the 1708 neutral turns in this set, evaluators were in complete agreement for only 340 turns, i.e. 19.9%. Whereas, across all emotions, human evaluators were in complete agreement with each other for only 2040 out of 5531 turns (36.88%). Given such ambiguity inherent in perceiving even acted expressions, the overall improvement in recall achieved here is of significant value.

Compared to IS09, RSM/sRSM performs slightly worse at recognizing anger. This can probably be attributed to the different frame-level features used across the two approaches. In this work, bag-of-words features are constructed from F0, energy and MFCCs. In contrast, the IS09 feature set uses two additional frame-level features: zero crossing rate (ZCR) and harmonic-to-noise (HNR) ratio. In [9], the same feature set combined with a tree-based classification scheme provided a similar recall for anger as IS09. This further suggests that the higher recall achieved on anger is mainly due



(a)

(b)

Figure 2.10: WA Recall vs. Dictionary Size for EMO-DB and USC IEMOCAP.

to the differences in LLDs and not due to different representations.

From an unsupervised RSM to an sRSM, a slight deterioration towards recognizing sadness is observed. This effect can be explained by the imbalance across different classes in the training examples. The cross-entropy error loss function aims to maximize the average classification accuracy, i.e. WA recall. Owing to the higher number of neutral and happy utterances, the topics learnt during the fine-tuning stage of sRSM are slightly biased towards these emotions. In order to address this issue, each class can be restricted to have the same number of examples during training, i.e. a balanced dataset. Alternatively, the loss function in Eq (2.3) can be modified to Eq (2.15), where, the training examples are individually weighted, $w^{(s)}$.

$$L(\theta) = -\frac{1}{S}\left[\sum_{s=1}^{S}\sum_{c=1}^{C} w^{(s)} 1\{t^{(s)} = c\} \log p(y^{(s)} = c | h^{(s)}, \theta)\right] + \frac{\lambda}{2}\|\theta\|_2^2 \qquad (2.15)$$

Classification performed directly over words obtained after VQ yields a WA recall of 51% for a dictionary of size $K = 512$. Recall purely based on chance is 30.88% for USC IEMOCAP, thus demonstrating the feasibility of a simple bag-of-words representation. Learning latent topics via LDA or RSM improves the performance significantly, as shown in Figure 2.10, mainly because they are able to identify repetitive, co-occurring patterns of words and yield a much simpler representation. In this regard, topic models can be viewed as an unsupervised dimensionality reduction technique applied over word counts [15].

2.6.3 SEMAINE

Cross-validation was not required for this database, since the training, development and test partitions, as specified in [62, 63], do not overlap in the speakers. The frame-level features are normalized as per the method outlined earlier for USC IEMOCAP. The development set was used to select the model parameters, i.e. the dictionary size, $K \in \{64, 128, 256\}$, and number of topics, $J \in [K/4, K/2]$. Results are reported for both partitions, development and test.

The results for arousal-based, binary classification and regression are shown in Table 2.7. As observed for the USC IEMOCAP database, LTMs outperform simple VQ-based features. Specifically, LDA and RSM achieve relative gains of 1.5% and 13.7%, respectively. Topics learnt in a supervised manner, as expected, lead to even further improvements; 7.6% and 14.6% for sLDA and sRSM, respectively. Compared to IS09, the proposed features demonstrate a significant improvement on the development set. Whereas on the test set, LDA and sLDA perform worse than IS09. RSM and sRSM are marginally better with relative gains of 0.7% and 1.4%, respectively. In case of regression, however, LTMs outperform IS09 on both the sets. Once again, sRSM yields the best performance with a COR of 0.384 and 0.444, compared to 0.238 and 0.288 using IS09, on the development and test sets respectively.

Table 2.8 shows a comparison for the case of valence-based, binary classification and regression. Compared to VQ, LDA and RSM demonstrate an improvement of

Table 2.7: Results for Arousal-Based Classification and Regression on SEMAINE. Classification Results are Expressed in Percentage (%). Symbols ∗ and † Indicate Statistical Significance over IS09 and VQ, Respectively.

Metric	IS09	VQ	LDA	sLDA	RSM	sRSM
			Development Set			
WA	60.73	60.72	63.85	65.31	66.04	**66.35**
UA	61.08	60.81	64.03	65.39†	66.02*†	**66.38***†
COR	0.238	0.325	0.350	0.364	0.357	**0.384**
			Test Set			
WA	67.16	66.86	67.90	71.03	71.47	**72.66**
UA	63.46	56.17	57.05	60.49	63.90†	**64.38**†
COR	0.288	0.255	0.312	0.322	0.430	**0.444**

8.8% and 10.3%, respectively. Once again, supervised learning via sLDA or sRSM improves upon its unsupervised counterparts. Unlike arousal, LTM-based features comprehensively outperform IS09 features. The latter, in this case, performs slightly worse than chance. Again, the best recall is obtained using sRSM - relative gains of 12.1% and 16.75% over IS09 on the development and test sets, respectively. In case of regression, sRSM obtains a COR of 0.349 and 0.171 on the development and test sets respectively, which is clearly better than 0.191 and 0.007 obtained using IS09.

The results obtained on SEMAINE are comparable to earlier works; in [74], a WA recall of 64.98% (arousal) and 63.51% (valence) was achieved using SVM and AdaBoost over statistics-based features. While, in [99], an UA recall of 65.7% (arousal) and 65.4% (valence) was achieved using a bag of HMMs approach. In each of these studies, the features were extracted over individual spoken words as opposed to turns, hence, a direct comparison is not feasible.

Based on the above experimental results on USC IEMOCAP and SEMAINE, the

Table 2.8: Results for Valence-Based Classification and Regression on SEMAINE. Classification Results are Expressed in Percentage (%). Symbols $*$ and $\dagger$ Indicate Statistical Significance over IS09 and VQ, Respectively.

Metric	IS09	VQ	LDA	sLDA	RSM	sRSM
	Development Set					
WA	59.61	58.33	63.03	64.10	65.50	**66.45**
UA	57.64	56.51	58.86	$61.96^{\dagger}$	$63.53^{*\dagger}$	**64.62***†
COR	0.191	0.191	0.330	0.332	0.327	**0.349**
	Test Set					
WA	49.93	51.56	55.13	57.21	56.32	**57.80**
UA	49.68	51.54	56.12^{*}	$57.63^{*\dagger}$	$56.88^{*\dagger}$	**58.00***†
COR	0.007	0.045	0.128	0.154	0.127	**0.171**

following observations can be made. Firstly, LTMs learn simplified, yet better, representations over acoustic words and their co-occurrences as demonstrated by the clearly higher recall obtained over VQ. Secondly, the performance difference between LDA and RSM can be attributed to the latter's distributed representations. In LDA, each word in a turn is assigned to a single topic, while, in RSM, each word is modeled by multiple topics. This allows each topic in the latter to define elementary features and their combination to give rise to more complex and richer representations. Combined with a lower complexity of inference, RSM-based approaches are more suited for tasks involving real-time recognition. Thirdly, learning topics in a supervised manner is highly beneficial, as evident from the improvements obtained using sRSM over competing LTMs on both databases. Finally, except for arousal-based binary classification on the test set of SEMAINE, each LTM outperforms turn-level statistics, i.e. IS09. These improvements are significantly higher for regression and valence-based classification over the spontaneous expressions of SEMAINE, suggesting that

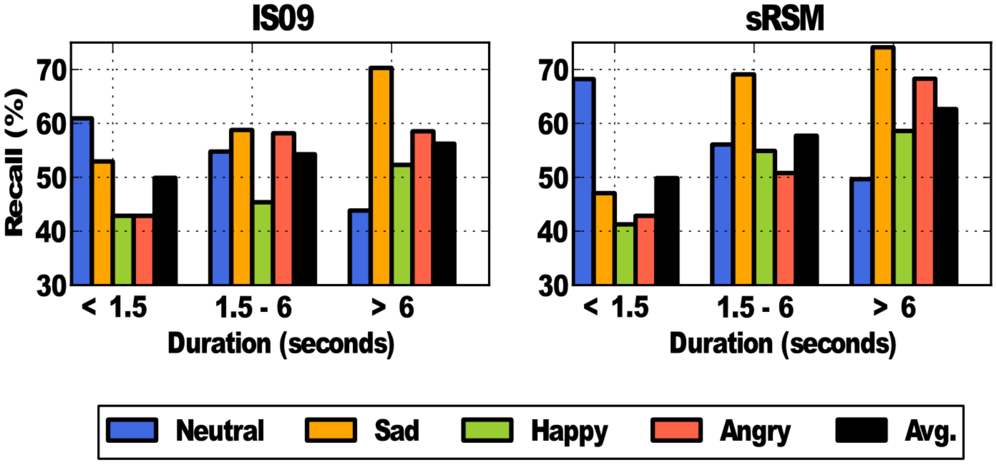


Figure 2.11: A Comparison Between the Recall Obtained Using IS09 and sRSM for Turns of Different Durations in USC IEMOCAP. Note the Increase and Relatively Better Performance of sRSM as the Duration of a Turn Increases from Less than 1.5s to Greater than 6s.

the co-occurrence information captured by the topics is highly representative of the underlying emotional content.

2.6.4 Effect of Turn Duration

As a result of the turn-based segmentation procedure, the duration of a turn varies depending on the speaker's activity. Turns are often long and and may consist of multiple emotions expressed in varying degrees and no seemingly regular structure. Consider, for example, neutral speech with occasional bursts of emotional activity. Experiments conducted to examine the behavior of LTM features with respect to the turn duration are presented below. First, all the turns are divided in three categories based on their duration: <1.5s, 1.5-6s and >6s. The UA recall over each category is used to compare the behavior of IS09 and sRSM-based features.

For USC IEMOCAP, there are 408, 3832 and 1291 turns in each category, respec-

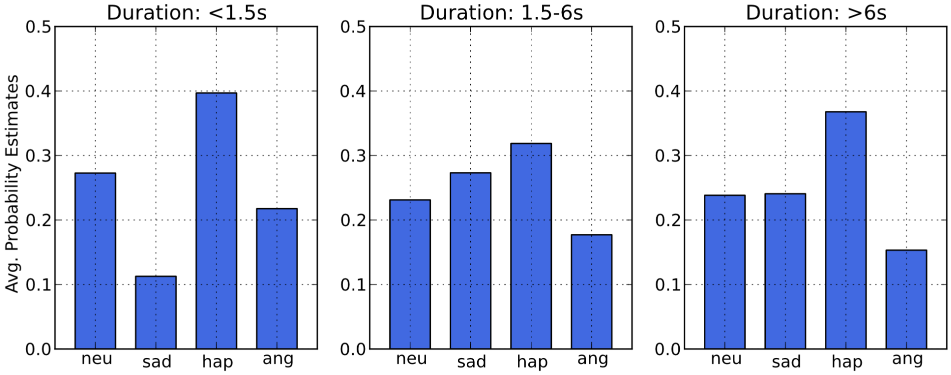


Figure 2.12: Error Analysis of Neutral Utterances Across Different Duration Categories Using sRSM. The Average Posterior Probability Estimates for All Misclassified Neutral Utterances in USC IEMOCAP Are Shown Here to Highlight the Ambiguity as the Turns Become Longer in Duration.

tively. The class-wise accuracy across the four emotion categories and their average is shown in Figure 2.11. The relative improvement from the shortest to the longest duration is 7.87% for IS09, while 14.38% for sRSM. The absolute difference in UA recall between sRSM and IS09 for turns less than 1.5s is -0.04%, whereas the difference for turns longer than 6s is 6.4%. Emotions such as sad, happy and angry are recognized with a higher accuracy as the duration increases, yet their accuracy is surprisingly low for shorter duration turns. This is probably due to the unavailability of enough sad/happy/angry examples with shorter durations. For instance, of the 408 turns with duration less than 1.5s, 47% are neutral.

The decline in recall rate of neutral speech, for either feature set, as the duration increases is particularly interesting, since a similar trend is not evident from the ground truth labels provided by human evaluators. The percentage of turns for which there is complete agreement for the three duration categories shows an increasing trend - 15.10%, 18.63% and 28.42%. Figure 2.12 shows the average posterior probability for all the misclassified, neutral utterances in USC IEMOCAP across the

Table 2.9: Effect of Turn Duration for Arousal-Based Classification on SEMAINE. Results are Expressed in Percentage (%). Symbol $*$ Indicates Statistical Significance over IS09.

Features	<1.5	1.5-6	>6
	Development Set		
IS09	54.78	51.41	62.85
sRSM	70.32^{*}	63.46^{*}	65.72
	Test Set		
IS09	54.93	60.65	76.31
sRSM	64.92^{*}	56.52	73.01

three duration categories. Such utterances tend to be misclassified as either happy or sad. Yet, the neutral content is captured as a secondary or minor emotion with slightly lower probability estimates. The emotional profile (EP) framework presented in [10] can be considered as a possible solution, which captures the major-minor emotions in order to resolve ambiguity. This framework is independent of the type of features or classifiers used and can be easily combined with the proposed approach.

For SEMAINE, there are 350, 373 and 237 turns in each duration category for the development set and 347, 219 and 107 for the test set. The results for arousal and valence classification are shown in Tables 2.9 and 2.10, respectively. For arousal, IS09 and sRSM are quite similar as one outperforms the other, either on the development set or the test set. On the other hand, sRSM achieves a significant gain over IS09 for valence discrimination - 26.9% and 35% on the development and test set, respectively.

Experimental results on both, USC IEMOCAP and SEMAINE, indicate that sRSM, and in general, LTMs are better suited to handle turns of longer durations. The extraction of turn-level statistics loses important local information, such as bursts of emotional activity, as the frame-level features are normalized over the turn. LTMs,

Table 2.10: Effect of Turn Duration for Valence-Based Classification on SEMAINE. Results are Expressed in Percentage (%). Symbol $*$ Indicates Statistical Significance over IS09.

Features	<1.5	1.5-6	>6
	Development Set		
IS09	53.56	52.53	52.74
sRSM	58.26	67.93*	66.95*
	Test Set		
IS09	50.50	53.09	43.96
sRSM	57.75	55.29	59.36*

in spite of generating turn-level descriptors, capture some local information from the word occurrences. The necessity for retaining such information is particularly relevant for valence-based discrimination, where sRSM demonstrates a significantly better recall over all turns and an even further improvement over turns longer than 6 seconds.

2.7 Cross-Corpus Evaluations

Speaker-independent, within-corpus evaluations are useful to provide a preliminary validation. However, real-world scenarios involve cases where the data does not belong to the same domain as the one used for training the system. For example, changes in elicitation techniques (acted vs. spontaneous), language, culture, accent, etc. are quite common. Cross-corpus evaluations, in such cases, can provide a more reliable measure of how well the approach generalizes across such differences.

The same databases, USC IEMOCAP and SEMAINE, are used. The data is first preprocessed to compensate for differences in recording conditions. Various methods, such as z-normalization [100] or min-max normalization [101] have been applied at the speaker and corpus level for this purpose. A corpus normalization approach is adopted in this work. Accordingly, the frame-level features of the training and test corpus are normalized to have the same mean. Accordingly, if M_{train} and M_{test} are the respective mean vectors of the training and test corpus, then each frame of the test corpus is multiplied by M_{train}/M_{test}. After normalization, acoustic words and topics are extracted using the dictionary and topic models estimated over the training corpus. Secondly, to ensure a valid comparison, the labels must be the same across each corpus. As described earlier, the turns of USC IEMOCAP were labeled categorically, while those of SEMAINE were labeled with binary arousal/valence attributes. The four categories of USC IEMOCAP are converted to binary, arousal (low - neutral/sad, high - happy/angry) and valence (negative - sad/angry, positive - neutral/happy) attributes.

The topics extracted over all turns of the training and test corpus are denoted as $\mathbf{h}^{trn}$ and $\mathbf{h}^{tst}$, respectively. The rows of $\mathbf{h}$ correspond to turns, while the columns to topics. For the test corpus, $\mathbf{h}^{tst}$ is further split in two disjoint partitions - (1) $\mathbf{h}^{tst,l}$, a set of turns with labels $y^{tst,l}$, and (2) $\mathbf{h}^{tst,u}$, the set of unlabeled turns used for evaluation. When SEMAINE is designated as the test corpus, $h^{tst,l}$ corresponds to the features extracted from the training set of SEMAINE, while $h^{tst,u}$ corresponds to the test set. Alternatively, when USC IEMOCAP is designated as the test corpus, $h^{tst,l}$ corresponds to a set comprising of 9 out of 10 speakers, while $h^{tst,u}$ corresponds to the remaining speaker. This process is repeated for each of the 10 speakers, resulting in a 10-fold process.

According to conventional cross-corpus experiments conducted earlier [100, 102], the test corpus is evaluated using the parameters, $\theta^{\star}$, of the classifier learnt over the training corpus as per Eq (2.16), where L is the cost function.

$$\theta^\star = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} L(y_i^{trn}, h_i^{trn}; \theta) + \frac{\lambda}{2} \|\theta\|_2^2 \tag{2.16}$$

These works do not account for the fact that emotions are perceived differently across geographical regions or cultures causing the annotations to be biased to their respective databases. In other words, even if the definition of labels are same across corpora, there is a significant difference between $p(y^{trn}|\mathbf{h}^{trn})$ and $p(y^{tst}|\mathbf{h}^{tst})$. Hence, the decision boundary learnt over the training corpus is no longer optimal for the test corpus. The results obtained in this case also indicate the joint performance loss due to both, the features and the classifier.

In order to improve the cross-corpus performance and determine the generalization of solely the topic features, two strategies are proposed to compensate for this bias. In each of these strategies, it is assumed that a few labeled turns from the test corpus are available, i.e. $\mathbf{h}^{tst,l}$, and that parameters $\hat{\theta}$ characterizing $p(y^{tst,l}|\mathbf{h}^{tst,l})$ can be learnt. Using $\hat{\theta}$ as a guide, new parameters are estimated from the training corpus such that the decision boundary changes to reflect the distribution of the test corpus.

2.7.1 Instance Selection

According to this strategy, instances in the training corpus that are not modeled well according to $p(y^{tst,l}|\mathbf{h}^{tst,l}, \hat{\theta})$ are identified. Such instances can be viewed as misleading or confusing, hence removing them would serve to bring $p(y^{trn}|\mathbf{h}^{trn}, \theta)$ closer to $p(y^{tst,l}|\mathbf{h}^{tst,l}, \hat{\theta})$. Accordingly, $\mathbf{h}^{trn}$ is first evaluated on $\hat{\theta}$. The the top k or all instances that are correctly classified are then selected and assigned a large weight, while a smaller weight is assigned to the wrongly classified instances. The new parameters $\theta^\star$ are now estimated via Eq (2.17).

$$\theta^{\star} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \alpha_i L(y_i^{trn}, h_i^{trn}; \theta) + \frac{\lambda}{2} \|\theta\|_2^2 \tag{2.17}$$

Here, α_i indicates the weight assigned to each instance. A simple rule is followed to set the weights: $\alpha_i = 1$ if correct, else $\alpha_i = 0$.

2.7.2 Weight Regularization

The differences between $p(y^{trn}|\mathbf{h}^{trn}, \theta)$ and $p(y^{tst,l}|\mathbf{h}^{tst,l}, \hat{\theta})$ can alternatively be explained by the difference in their weights θ and $\hat{\theta}$. In traditional $L2$-norm regularization, i.e Eqs (2.16) and (2.17), the weights are penalized from becoming too large. If instead, the difference, $\|\hat{\theta} - \theta\|_2^2$, is penalized from being large, then the new weights will be such that $\theta \rightarrow \hat{\theta}$. Parameters $\theta^{\star}$, in this case, are learnt as per Eq (2.18).

$$\theta^{\star} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} L(y_i^{trn}, h_i^{trn}; \theta) + \frac{\lambda}{2} \|\hat{\theta} - \theta\|_2^2 \tag{2.18}$$

2.7.3 Experimental Results

The best within-corpus (WC) results are obtained from the experiments described earlier. In addition to different methods used to elicit emotions, USC IEMOCAP and SEMAINE are also recorded using subjects belonging to different cultures. The former comprises of American speakers, while the latter comprises of speakers from 8 countries across Europe. These factors affect the performance such that the cross-corpus recall will, in general, be lower than under a within-corpus setting [100].

The UA recall obtained using a conventional cross-corpus strategy without adaptation for different LTMs and train/test scenarios are shown in Figure 2.13. For LDA, sLDA, RSM and sRSM, the average deterioration across all the scenarios is 11.1±3.3%, 6.6±3.1%, 8.7±3.3% and 5.2±3.4%, respectively. The recall values for cross-corpus using instance selection are presented in Figure 2.14. In this case, the

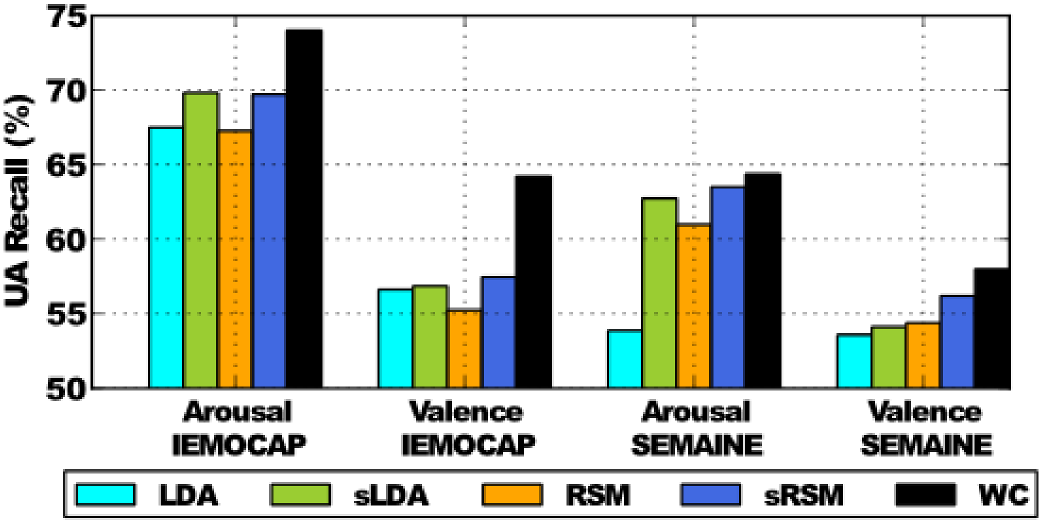


Figure 2.13: Cross-Corpus Recall Without Adaptation. Horizontal Axis Indicates the Classification Task and Test Corpus. The Figure Shows a Detailed Comparison Between Four Different LTMs along with the Best Within-corpus (WC) Recall in Each Case.

average deterioration across all the scenarios is 8.1±1.7%, 5.1±1.7%, 5.3±2.4% and 2.6±1.8% for LDA, sLDA, RSM and sRSM, respectively. Similar results for cross-corpus using weight regularization are shown in Figure 2.15. The average deterioration across all the scenarios is 7.5±4.1%, 5.8±3.6%, 5.3±3.9% and 2.7±2.3% for LDA, sLDA, RSM and sRSM, respectively.

The improvements demonstrated by either adaptation strategy over a conventional approach confirm the existence of a classifier-specific bias due to varying perceptions across corpora. Adaptation successfully reduces this bias by using parameters ($\hat{\theta}$) as a reference during learning. Between the two approaches, the mean deterioration is almost similar for both instance selection and weight regularization, however, the latter has a comparatively larger standard deviation, thus making the former a more suitable approach. Experiments were conducted to combine the two strategies, which

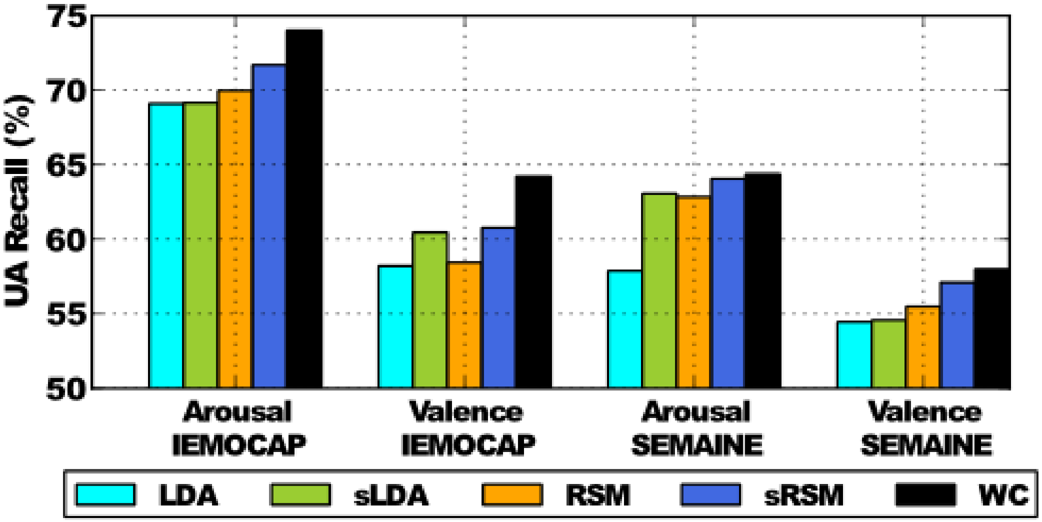


Figure 2.14: Cross-Corpus Recall with Instance Selection. Horizontal Axis Indicates the Classification Task and Test Corpus. The Figure Shows a Detailed Comparison Between Four Different LTMs along with the Best Within-corpus (WC) Recall in Each Case. There Is a Significant Improvement in Performance as Opposed to a Conventional Approach Without Instance Selection, i.e. Figure 2.13.

did not yield any significant improvements.

When the spontaneous expressions of SEMAINE are evaluated over the acted expressions of USC IEMOCAP, the relative deterioration using sRSM with instance selection and weight regularization is 1.0±0.5% and 1.1±0.8%, respectively, compared to 4.2±1.1% and 4.4±1.9%, respectively, for the vice versa case. This can mainly be attributed to the number of examples available for training the classifier; USC IEMOCAP is approximately 5 times larger than SEMAINE. Between arousal and valence-based classification, the deterioration is more severe for the latter case. Using sRSM with weight regularization and instance selection, a relative deterioration of 1.8±1.3% and 1.4±1.1%, respectively, is obtained for arousal. Whereas, a relative deterioration of 3.5±1.9% and 4.2±2.3%, respectively, is obtained for valence. Again, the inherent limitations of speech coupled with the differing perceptions of valence

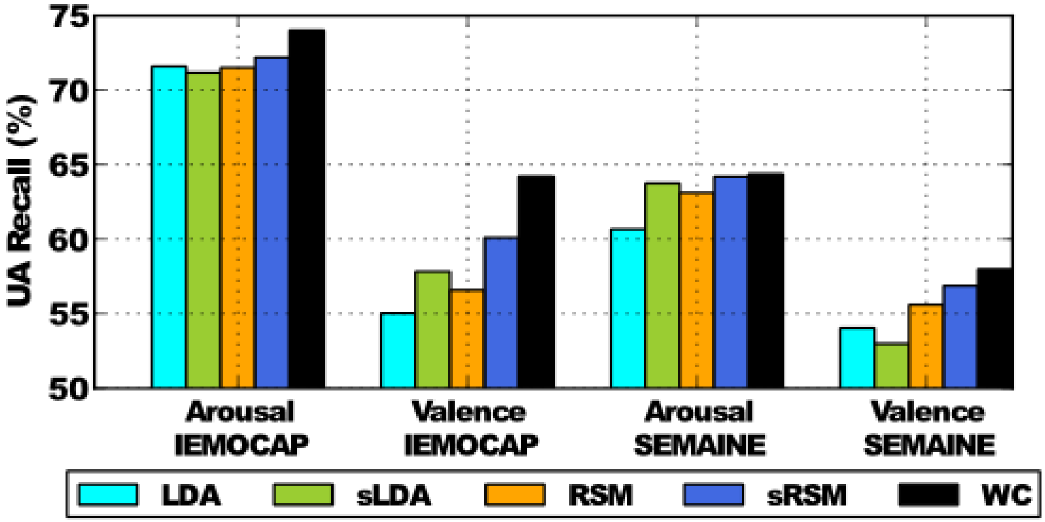


Figure 2.15: Cross-Corpus Recall with Weight Regularization. Horizontal Axis Indicates the Classification Task and Test Corpus. The Figure Shows a Detailed Comparison Between Four Different LTMs along with the Best Within-corpus (WC) Recall in Each Case. There Is a Significant Improvement in Performance as Opposed to a Conventional Approach with Standard L2-Regularization, i.e. Figure 2.13.

across cultures and geographical regions possibly account for this loss. This phenomenon was also observed in a previous cross-corpus study conducted over different databases [100]. There are no previous reports of cross-corpus studies over the two databases used in this study.

Between different LTMs, the supervised LTMs outperform their unsupervised counterparts as observed in a within-corpus setting. sRSM, once again, achieves the least deterioration across all train/test scenarios and adaptation approaches. In case of LDA and RSM, the topics learnt initially over the training corpus remain unchanged and only the top-level classifier is modified. Although the results show a relatively higher deterioration compared to sRSM or sLDA, they single out the performance loss due to the topic-based features alone and are indeed promising.

Table 2.11: Comparison of Software Implementation for Different Approaches.

Technique	**Computation time (ms)**	
	Feature Extraction	Classification
HMM	61.56	2.14
IS09	81.06	0.03
LDA, K=64	63.44	1.13
LDA, K=512	64.98	6.16
RSM, K=64	63.44	0.02
RSM, K=512	64.98	0.17

2.8 Software Implementation

The feature extraction, post-processing, LDA and RSM routines are implemented on a Lenovo laptop with an Intel i7 2.7 GHz quad-core processor and 4 GB RAM. OpenMP [103], a freely available software for parallel computing, is used to optimize LDA and achieve speed-up by a factor of 10. Table 2.11 highlights the implementation complexity measured by the time taken to process a turn of 1 second duration. Inference in LDA requires $O(NJ)$ operations per iteration, which accounts for the higher recognition time in comparison to RSM, where topics are inferred in a single pass, i.e. a total complexity of $O(KJ)$. Either topic model requires less time compared to the IS09 approach, where the extraction of statistical features incurs additional complexity.

2.9 FPGA Implementation

As the size of a database grows, K and J will grow accordingly, thus increasing the processing time. An FPGA-based parallel implementation for LDA is considered in order to improve the real-time performance. The hardware architecture for the proposed algorithm consists of two main blocks: feature extraction and LDA inference. Both blocks are implemented using Verilog HDL and synthesized on Xilinx Virtex-5 device (XC5VSX240T). The design is verified using Modelsim. For word lengths of 16, 20 and 24 bits, the corresponding classification error rates are 2%, 0.5% and 0% respectively. Based on these results, a 24-bit fixed-point representation is chosen for the FPGA implementation.

A summary of the FPGA resource utilization for feature extraction is presented in Table 2.12. The FIR filter and FFT routines are implemented using Xilinx IP cores. The Mel-bank transform and DCT multiplications are implemented using DSP slices. The utilization is fairly low (8%). A pipelined implementation of feature extraction for 25ms of speech (1 word) takes 841 cycles, while post-processing takes 265 cycles. With a system clock rate of 100 MHz, the total processing period for feature extraction from 1.5s of speech (150 words) is T_{fe}=8.41μs×150 = 1.261ms and that of post-processing is T_{pp}=2.65μs×150 = 0.397ms.

The LDA inference algorithm described in Figure 2.3 is implemented by a multiple processing element (PE) architecture, where each PE is assigned to one topic. For the 60-topic system studied here, there are 60 PEs. The critical step of LDA inference is to obtain the digamma factor $\Psi(\gamma)$. One straightforward method is to use a look up table (LUT). In this study, $\gamma \in [0.0001, 500]$ and for a resolution of 2^{-14}(<0.0001), the size of LUT is 500×2^{14}=8MB. This exceeds the storage space available on the FPGA chip. Alternatively, a Taylor series approximation can also be used. Since the digamma computations are identical for all topics, only one such calculation engine is implemented and placed in a central unit (CU). To avoid access conflicts, the PEs implemented in a staggered fashion. The other computations in each PE, such as division and exponential functions, are implemented using Xilinx CORDIC IP core

Table 2.12: Resource Utilization for Feature Extraction.

Unit	Occupied slices	Slice Reg.	Slice LUTs	Block RAM	DSP
FIR	153	163	111	0	1
Ham	0	0	1	1	1
FFT	1729	1854	1723	3	10
Mel	897	960	6400	1	40
DCT	268	288	1920	1	12
Total	3047 (8%)	3265 (2%)	10155 (6%)	6 (1%)	64 (6%)

Table 2.13: Resource Utilization for LDA Inference Engine.

Unit	Occupied slices	Slice Reg.	Slice LUTs	Block RAM	DSP
PE	317	1057	977	1	1
CU	4183	13925	7468	1	9
Total	23203 (62%)	77345 (52%)	66088 (44%)	61 (12%)	69 (7%)

and multiplications are implemented using DSP slices.

FPGA resource utilization for the LDA inference engine is presented in Table 2.13. Each PE occupies 317 (0.8%) slices and CU occupies 4,183 (11%) slices, thus the total occupied slices is 23,203 (62%). For an utterance with 150 words, one LDA iteration takes 295 cycles. Thus, the total processing period for LDA inference (50 iterations) is T_{LDA}=2.95μs×50=0.147ms. Finally, the total processing time for an utterance of duration 1.5s is $T_{fe} + T_{pp} + T_{LDA}$=1.805ms.

While post-processing is relatively higher compared to the software counterpart, the time complexity for LDA is reduced due to implementation of topic-level parallelization. Similarly, the time complexity of feature extraction is reduced because FPGA efficiently utilizes the parallelism in the component algorithms. For large databases, K and J will be larger. While the feature extraction time will remain the same, the classification time will increase linearly. For fixed J, as K increases, the time for post-processing and LDA will increase linearly. For fixed K, as J increases, only the LDA processing time increases linearly, others are unchanged. Finally, even for large K=1024 and J=180, the processing time of the FPGA based system is estimated to be only 2.65ms.

2.10 Summary

In this work, a novel approach for the extraction of turn-level features using LTMs was presented. Parallels are drawn between text documents and emotional speech; the latter can be viewed as a mixture of multiple emotion-specific topics, where, the topics captures salient information from the co-occurrence patterns of LLDs. Two fundamentally different models, LDA and RSM, and their supervised counterparts were considered for the purpose of learning features. Furthermore, sRSM, which treats the RSM as a pre-training stage followed by fine-tuning via backpropagation, was proposed to learn features that are optimal for discriminative tasks.

The derived features were evaluated on different types of emotional expressions and output representations and were shown to outperform state-of-the-art methods in each case. On the acted emotions of USC IEMOCAP, sRSM obtained a relative improvement of 7% compared to turn-level statistics collected by a brute-force methods. Whereas on the spontaneous expressions of SEMAINE, sRSM obtained an improvement of 16.75% for valence-based classification, which is quite significant considering the well-known difficulty of valence discrimination using only speech information. With respect to the turn duration, sRSM and in general, LTMs, were shown to be better suited for longer turns (>6s), which is strongly desirable for current turn-based practices. The improvement over turn-level statistics for valence-based classification is particularly significant, 26% and 35% on the development and test sets of SEMAINE, respectively.

In a cross-corpus setting, it was shown that classifiers are inherently biased because of the annotation procedures and cultural perceptions specific to each corpus, which leads to poor generalization. To compensate for this bias and improve cross-corpus performance, two novel adaptation strategies were proposed. Compared to the best within-corpus performance, sRSM showed the least relative deterioration of only 2.6% and 2.7% using instance selection and weight regularization, respectively. This further highlights that the proposed approach can efficiently generalize across different accents, speakers and elicitation types (acted vs. spontaneous).

Qualitative aspects of the features were investigated using a normalized pointwise mutual information measure between topics and emotions. Analyses revealed the emotions to be naturally and well separated in the topic space. This shows that the co-occurrence information captured by topics is strongly related to the underlying emotion, thus offering a novel, generative-model based interpretation of how emotions influence the observed speech characteristics.

Software implementation complexity was assessed to determine the feasibility for real-time emotion recognition. Furthermore, an FPGA-based implementation of an LDA-based framework with a dictionary size of 512 and 60 topics was developed and was able to identify emotions in an utterance of duration 1.5s in 1.8ms.

Finally, a short comment on the flexibility of the proposed approach. Although energy, F0 and MFCCs were used as frame-level features in this work, words and topics can be derived from other frame-level features or modalities and be combined to decrease the confusion between happy-angry and neutral-sad emotions, and lead to further improvements. Similarly, a simple logistic/softmax regression classifier can be replaced by more sophisticated classifiers [73] or alternative tree-based schemes [9] to achieve even better discrimination.

Chapter 3

MULTI-MODAL EMOTION RECOGNITION

In Chapter 2, an LTM-based approach was presented for speech-based emotion recognition. Features were extracted from low-level, acoustic descriptors. Other sources including facial expressions, language or physiological processes have also been considered for interpreting human emotions [104, 105]. Individually, the sources offer limited insight, but, their combination provides a better understanding of the context, and, consequently the speaker's emotional state [8].

In this chapter, a multi-modal emotion recognition framework using undirected topic models is proposed. The model, RSM [16], described in Chapter 2 is extended to perform feature extraction from multiple modalities - facial expressions, speech and language. Recall that, according to RSM, each document (turn) is represented as a mixture of topics (emotions), and each emotion-specific topic is a discrete distribution over words (low-level descriptors) in a dictionary. Classification is then performed over latent topics inferred from the observed words. In addition to RSM, LDA was also shown to yield reasonable success. However, the recognition and training process was very slow, which makes RSM a more viable alternative. In a topic model, temporal information is ignored, similar to conventional supra-segmental methods. Yet, unlike them, it successfully captures the complex variations without resorting to a brute-force collection of statistical features.

3.1 Proposed Approach

The proposed approach, starting from low-level feature extraction to inferring latent topics and classification, is described in this section.

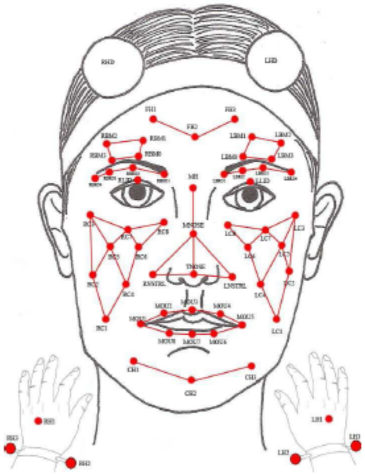


Figure 3.1: Arrangement of Face Markers in the USC IEMOCAP Database.

3.1.1 Bag-of-Words Features

Acoustic, low-level descriptors include prosodic and spectral features, such as pitch, energy and Mel frequency cepstral coefficients (MFCCs). These descriptors are extracted on a frame basis, at a rate of 100 frames/second. A turn is thus represented by a sequence of multi-dimensional, real-valued features. In order to be relevant for topic models, these features are transformed to discrete symbols, analogous to words in a document. Hence, a dictionary of candidate feature vectors is learnt via VQ. Each vector is mapped to the index of the dictionary vector it is closest to based on the Euclidean distance. The size of a dictionary is dependent on the data and it ranges from 64 to 512. Further details of this extraction procedure were presented in Chapter 2.

Facial expressions are characterized using multiple facial markers, an example arrangement of which is shown in Figure 3.1. Each marker is denoted by its (x, y, z) co-ordinates and the nose marker is defined as the local co-ordinate center of each frame. Excluding the nose, head and hand markers, the remaining 46 markers (138 co-ordinates) are divided into 3 distinct regions - lower, middle and upper. The lower region includes 11 chin and mouth markers, the middle region comprises of 16 left/right cheek markers, and the upper region includes 19 left/right eyebrow and forehead markers. Face normalization is performed as per the method outlined in [11]. Region-specific principal component analyses (PCA) is further applied and the top 13, 16 and 20 eigenvectors are retained for the lower, middle and upper regions, respectively. Similar to the process carried out for acoustic descriptors, VQ is applied to generate bag-of-words features with the dictionary size ranging from 32 to 128 for each region.

Language-specific features can be constructed using the results obtained from a speech recognition system. In this study, a perfect speech recognizer is assumed and transcripts provided with the database are used to extract word information. Stemming and stop-word removal techniques are used to preprocess the transcripts, resulting in a dictionary with 2500 distinct words. Of these, only the 500 most frequently occurring words are retained.

3.1.2 Classification

A topic model learnt for each source of information results in a set of 5 models corresponding to the 3 visual regions, 1 speech and 1 language features. The latent topics h are inferred from the observations v in a single pass via Eq (2.7). The expectation is that turns with different emotions will have a dissimilar distribution of topics in contrast to the scenario where they belong to the same class. In this regard, the purpose of a topic model is to learn a simplified, intermediate representation over noisy, low-level bag-of-words features in an unsupervised fashion. An SVM classifier trained over the topics is then used to perform classification and to assign a label.

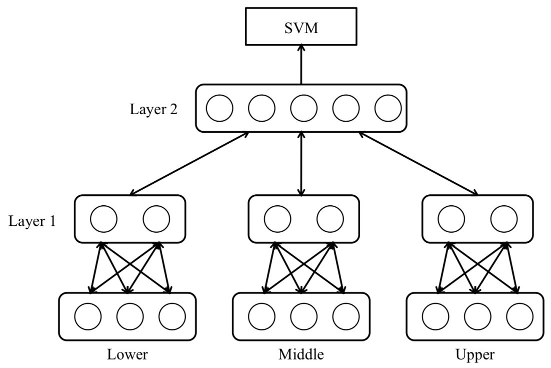


Figure 3.2: Feature-Level Fusion for Combining Features from 3 Face Regions.

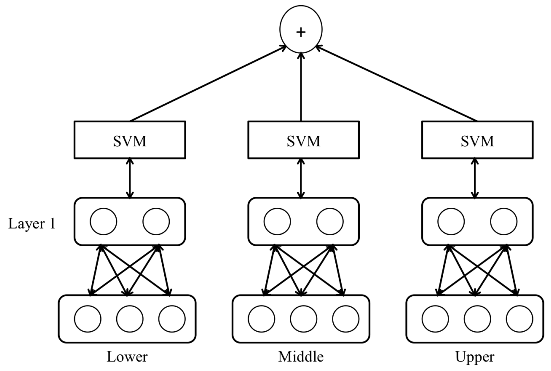


Figure 3.3: Decision-Level Fusion for Combining Features from 3 Face Regions.

Multiple strategies are considered for the fusion of facial, region-specific topics. Feature-level fusion is shown in Figure 3.2. An additional layer of features is learnt over the topics combined from the 3 face regions using a simple RBM. Alternatively, as shown in Figure 3.3, a decision-level fusion approach arrives at a final decision by weighting the outcome from classifiers trained separately over each region.

For multi-modal classification, a decision-level fusion approach is employed, where the classifier estimates from the face, speech and language topics are weighted equally. Feature-level fusion is ill-suited here since the number of weights to be learnt increases significantly (from 0.1 million to 1 million), requiring more training data.

3.2 Database

The USC IEMOCAP [97] database, also described earlier in Chapter 2, comprises of acted conversations between 5 male-female actor pairs. Facial expressions captured in the form of motion markers placed at different points on a speaker's face are provided with the database. Following the procedure described in Chapter 2, only prototypical turns that receive majority consensus among evaluators, are selected. Furthermore, turns labeled as neutral, sad, happy or angry are retained. Following earlier approaches [9, 10], happy and excitement are treated as the same emotion. Of the 5531 turns, facial marker information is provided for approximately half of the turns. The distribution of turns for the database is as follows: 606 neutral (N), 653 sad (S), 882 happy (H), 621 angry (A), i.e. a total of 2762 turns.

3.3 Experimental Results

In order to ensure speaker independence, experiments are performed using a Leave-One-Speaker-Out (LOSO) strategy. This results in a 10-fold process, with a separate RSM and SVM classifier trained for each fold. Unweighted average recall (UA), as defined in the InterSpeech Emotion Recognition challenge [61] and Chapter 2, denotes the average class-wise accuracy and is used as the metric to evaluate performance.

Each source and region-specific RSM is trained using stochastic gradient descent with a batchsize of 100 samples. The learning rate and momentum are fixed at 0.002 and 0.8 respectively. The number of hidden topics being dependent on the data, vary for each fold and a search is through $[K/4, K]$, combined with cross-validation over the training set, is used to empirically determine the optimal value. A similar procedure is followed for determining the optimal dictionary size for each modality. A linear kernel, multi-class SVM is used for classification over the inferred topics.

Among the 3 face regions, the lower region of the face shows the best results with an UA recall of 59.43%. The middle and upper regions yield a recall of 53.05% and 51.52%, respectively. The superior performance of the lower region can be attributed to the strong correlation that exists between the expression of emotions and lip/mouth movements. Between a feature-level and decision-level fusion of these regions, the former shows a slightly better performance with a recall of 60.71% compared to 59.11% for the latter. Either approach comprehensively outperforms the best-known result of 55.74% in [11], which uses emotion-specific HMMs.

Further inferences can be drawn from the confusion matrix shown in Table 3.1. It is evident that face expressions are particularly suited for recognizing happy emotions at a high rate, i.e. 81.85%. The respective accuracies for neutral, sad and angry are 38.94%, 53.44% and 68.59%. In contrast, the approach in [11] reports a class-wise accuracy of 34.79% (N), 53.68% (S), 76.98% (H) and 57.52% (A), which is clearly inferior to the proposed approach. The problem of recognizing neutral emotions is challenging since multiple definitions of neutrality exist based on the speaker's context.

From Table 3.1 (b), it is clear that speech is strong at recognizing sadness, with an accuracy of 77.64%. Similarly, from Table 3.1 (c), language is best at classifying neutral emotions, with an accuracy of 68.48%. Thus, when considering diverse sources, it is expected that the best characteristics of each source will be retained. The confusion matrix for multi-modal fusion, Table 3.1 (d), demonstrates a class-wise accuracy of 64.52% (N), 68.75% (S), 78.79% (H) and 63.60% (A). The UA recall is 68.92% for this combination; there is no previous work that reports results for a combination of these sources on this database. Furthermore, fusion also compensates for the deficiencies of each source. For example, face or language features exhibit a high confusion between sad and neutral emotions, which is better discriminated by con-

Table 3.1: Confusion Matrix for Different Sources.

	N	S	H	A
N	**236**	141	94	135
S	132	**349**	90	82
H	25	31	**722**	104
A	69	41	85	**426**

(a) Face, Feature-level fusion

	N	S	H	A
N	**255**	151	147	53
S	63	**507**	59	24
H	147	123	**486**	126
A	87	44	150	**340**

(b) Speech

	N	S	H	A
N	**415**	46	91	54
S	198	**282**	111	62
H	251	71	**524**	36
A	207	48	86	**280**

(c) Language

	N	S	H	A
N	**391**	79	80	56
S	124	**449**	54	26
H	109	37	**695**	41
A	116	20	90	**395**

(d) Multi-Modal

sidering speech features; or, the high misclassification rate between happy and angry emotions in speech is compensated by the considering face or language information.

The performance for each source and their combination is summarized in Table 3.2. Recall for speech is 57.39%, compared to a decision tree based performance of 58.46% reported in [9], while a combination of face and speech yields a recall of 66.05% compared to the deep belief network performance of 66.17% reported in [10]. These methods followed a conventional supra-segmental approach based on a brute-force collection of statistics. Although the size of the dataset used here is slightly different from these methods, these comparisons indicate that LTM-based features can yield comparable results.

Table 3.2: Recognition Performance for Each Source and Their Combinations.

Source	**UA Recall (%)**
Face	60.71
Speech	57.39
Language	54.04
Face + Speech	66.05
Face + Language	64.24
Speech + Language	61.96
Face + Speech + Language	68.92

3.4 Software Implementation

The feasibility of this multi-modal approach for real-time emotion recognition is determined by profiling its software implementation. Each source is processed sequentially on a Lenovo laptop with an Intel i7 2.7 GHz quad-core processor and 4 GB RAM. The average classification time for a turn of 1 second duration is shown in Table 3.3.

Results indicate that for a combination of all 3 sources, the classification time is approximately 666.65ms. It is also evident that most of the time is spent in preprocessing and feature extraction. In case of speech, the computation of MFCCs via FFT accounts for most of the time [17]; in case of facial expressions, the translation and rotation operations applied to the markers accounts for most of the time [97]. The estimates presented here also include the time taken for speech recognition, which is necessary to perform language-based recognition and takes approximately 600ms. Although, an implementation of the same is not openly available, the results are obtained from recently published benchmarks [83], which uses the popular DBN/HMM based framework. Furthermore, in this study, the facial expression infor-

Table 3.3: Implementation Time (ms) for Each Source.

	Face	**Speech**	**Language**	**Multi-Modal**
Preprocessing + Feature Extraction	26.46	64.98	600	665.44
RSM + SVM	0.67	0.20	0.34	1.21
Total	27.13	65.18	600.34	666.65

mation is readily available via markers; an extraction of this information in real-world scenarios would require computationally intensive image processing techniques [8].

3.5 Summary

In this chapter, LTM-based features were extended to perform emotion recognition from multiple modalities. Results indicate that topic models are well-suited to capture the complex variations exhibited in speech, language and facial expressions. Multiple strategies, feature and decision-level fusion, were presented to classify facial expressions. Using the former, a relative improvement of 8.89% was achieved over state-of-the-art methods. A comparable performance was also achieved for speech-only or a combination of both facial and speech information. The 3 sources were individually identified to perform best at recognizing happy (face), sad (speech) and neutral (language) emotions, while their fusion was shown to retain these characteristics and increase the average class-wise accuracy by a significant margin to 68.92%. Via software implementation, the classification time for a turn of 1 second duration was estimated to be 666.65ms, which ensures that the proposed framework satisfies real-time requirements.

Chapter 4

ARTICULATION CONSTRAINED LEARNING

Besides the acoustic characteristics and spoken content, emotional speech can also be characterized by articulatory kinematics. A majority of the research in this area is focussed on using the acoustic properties of speech, owing to their strong correlations with emotion and simple recording procedures. However, it is commonly understood that speech articulation also exhibits a strong correlation with emotion. One such example highlighting this relationship between emotions, articulatory movement and acoustic characteristics is depicted in Figure 4.1. Here, for the vowel */AE/* in the word *compare*, anger forces a larger opening of the jaw as opposed to sadness. Similarly, the lip protrusion is more towards the outside for the vowel */IY/* in the word *me* under anger. The differences in articulatory movement correspond to distinct behaviors in the frequency domain between anger and sadness for these particular vowel segments. Hence, methods that exploit this strong correlation between acoustic and articulatory data could potentially yield more accurate and reliable emotion recognition systems.

There are relatively very few studies that attempt to characterize emotions using articulatory information. In [20], it was shown that the degree of jaw opening increased significantly as subjects became annoyed (or irritated), while, in [21], the lateral lip distance between the corners of the mouth was shown to be strongly influenced by the emotional state. In [22], the authors showed that articulation-based features achieved a much better classification rate compared to acoustic features for a single male subject. Articulatory data in each of the above works was collected using an electromagnetic articulography (EMA) system consisting of sensors attached to various locations on a subject's mouth. Alternatively, articulatory data captured us-

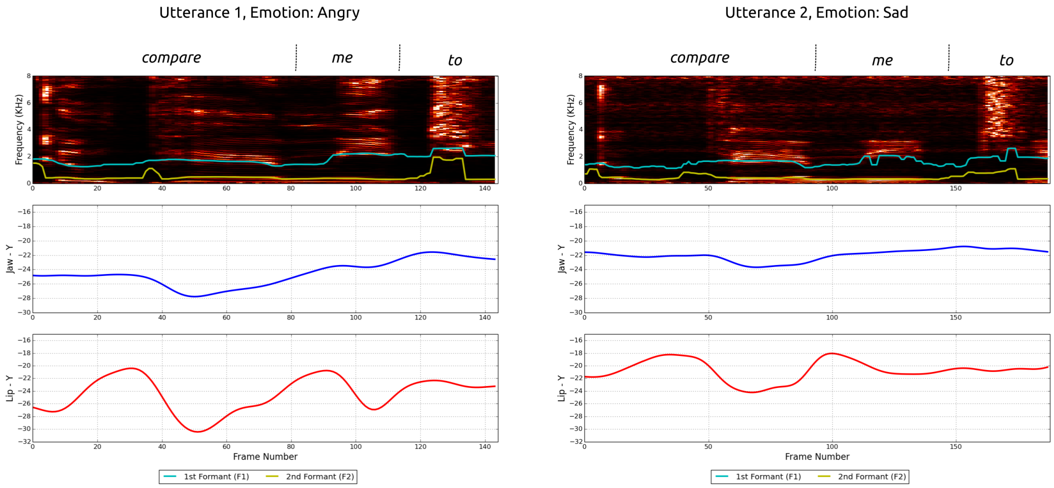


Figure 4.1: Relationship Between Emotions, Acoustic Characteristics and Articulatory Information for an Utterance *"compare me to"* by a Male Speaker. (Top) Spectrogram and Formant Tracks, (Middle) Position of the Jaw along Y-axis, and (Bottom) Position of the Lip along Y-axis. Note the Differences in Articulatory Position and Frequency Response for Vowels */AE/* and */IY/* of Words *"compare"* and *"me"*, Respectively. Negative Axis Corresponds to Downward Movement along the Y-axis.

ing facial markers, have been applied in a multi-modal framework [11, 18]. In [18], the authors showed that the lower region of the face (chin and lips) was the best indicator of emotion. These studies are mostly limited to single subjects, or multiple speakers recorded under similar conditions. However, more importantly, these methods require articulatory data to be available during the recognition step in order to perform reliably. Acquisition of such data on a large scale is difficult and time-consuming due to its invasive and highly sensitive recording procedure, which limits the scope and application of these methods to only laboratory environments.

In this chapter, a novel, discriminative learning method for emotion recognition using articulatory and acoustic information is proposed. The advantage of this method is that articulatory data is required only during the training step, thus overcoming the limitations of aforementioned studies. The proposed articulation constrained learning (ACL) method is set up to jointly minimize emotion classification error and articulatory reconstruction error, using acoustic features from the same or different domains. Specifically, a conventional logistic regression cost function is extended to include additional constraints that enforce the model to also reconstruct articulatory data. The classifier weights are constrained to be sparse via L1-regularization, which leads to a shared and interpretable representation. The proposed method improves the generalization ability of the classifier to work better on unseen samples. Furthermore, ACL is well suited for databases with high dimensional feature sets and limited articulatory data, as commonly found in emotion recognition studies.

The remainder of this chapter is organized as follows: The databases used in this work are described in Section 4.1. The proposed ACL method is described in Section 4.2. Experiments and results for within and cross-corpus scenarios are presented in Sections 4.3 and 4.4, respectively.

4.1 Data Preparation

A brief overview of the two databases, USC EMA and USC IEMOCAP, and the respective articulatory information is described in this section.

4.1.1 USC EMA

This database [22] comprises of scripted and acted emotions by three (1 male, 2 female) speakers. A set of 14 sentences, mostly neutral in emotional content, were used. Four different emotions, i.e., neutral, angry, sad and happy, were simulated by each speaker. The male speaker recorded each sentence 5 times for each emotion, resulting in a total of 280 utterances. The female speakers performed the same exercise, with only 10 out of 14 sentences, resulting in 200 utterances per speaker. From a total of 680 utterances, only those utterances were chosen for which external evaluators were in consensus with regards to the perceived emotion, resulting in a set of 503 utterances.

Articulatory data was collected using an EMA system. The positions of three sensors attached to the tongue tip, the lower maxilla (for the jaw movement) and the lower lip were tracked. Each sensor trajectory (target) was recorded in the x-direction (forward-backward movement) and the y-direction (vertical movement). Along with the position, velocity and acceleration of each trajectory are also included. This resulted in a total of 18 articulatory targets, as shown in Table 4.1. Articulatory data was recorded at a sampling rate of 200 Hz, while speech was recorded at a sampling rate of 16 KHz.

Figure 4.2 shows the mean value of the X, Y coordinates for the tongue, jaw and lip positions for different vowels and emotions. Angry utterances show the most distinct characteristics compared to other emotions. This effect is prominent especially for the vowels */AE/* and */AA/*. Previously, researchers studied these aspects in detail and showed that such differences are statistically significant [22]. This further highlights the dependence between emotions and articulatory kinematics.

4.1.2 USC IEMOCAP

The USC IEMOCAP database [97] was collected by asking five pairs of male-female actors to elicit emotions either by reading from a script or via improvisation in a conversational setting. This database consists of a total of of 10,039 utterances. Categorical attributes, including neutral, sad, happy, angry, frustrated, surprised, disgust, fear, and unknown, are assigned to each utterance. Only scripted utterances for which a majority consensus was reached among external evaluators are considered in this study. Further, utterances labeled as neutral, sad, happy, and angry are selected, while the remaining attributes are not considered as they are under-represented. This

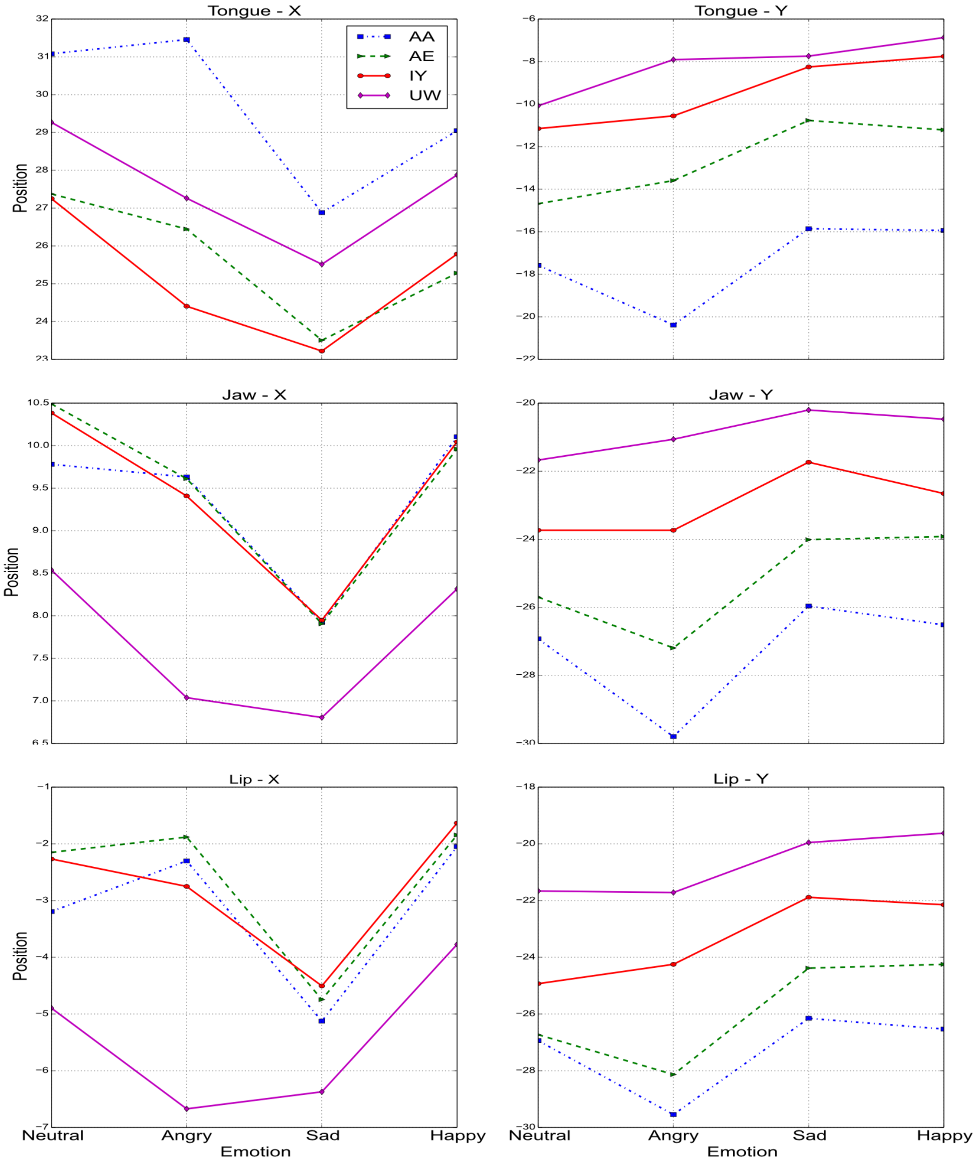


Figure 4.2: Mean Position of Select Articulatory Targets Across Different Vowels and Emotions in USC EMA. (Top) Tongue, (Middle) Jaw, and (Bottom) Lip. Negative Axis Corresponds to Forward Movement along X-axis and Downward Movement along Y-axis.

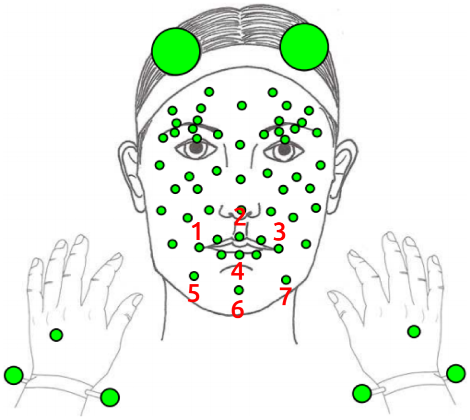


Figure 4.3: Arrangement of Facial Markers in USC IEMOCAP. The Markers Used in This Study Are Numbered from 1 to 7.

results in a total of 1262 utterances distributed across ten speakers and four emotions.

Here, articulatory information is available in the form of motion capture markers located at different points on a speaker's face. An example arrangement showing 53 facial markers is shown in Figure 4.3. These markers were originally intended for studying facial expressions, hence, not all markers contain information relevant to articulation. Markers located in the chin and lip areas are considered in this study. Specifically, the chin position (6), width of the chin (difference between 5 and 7), lower lip position (4), lip height (difference between 2 and 4), and lip width (difference between 1 and 3) are considered. Each marker is represented by its (x, y, z) co-ordinates, resulting in a total of 15 articulatory targets, as described in Table 4.1.

Figure 4.4 shows the mean value of the X, Y, Z coordinates for the chin and lower lip positions for different vowels and emotions in the USC IEMOCAP database. There is a strong correlation between the chip and lip positions along the Y, Z axes. Similar to USC EMA, the articulatory behavior is different across emotions for the same vowel. For instance, observe the chin movement along the Y axis for the vowel $/UW/$.

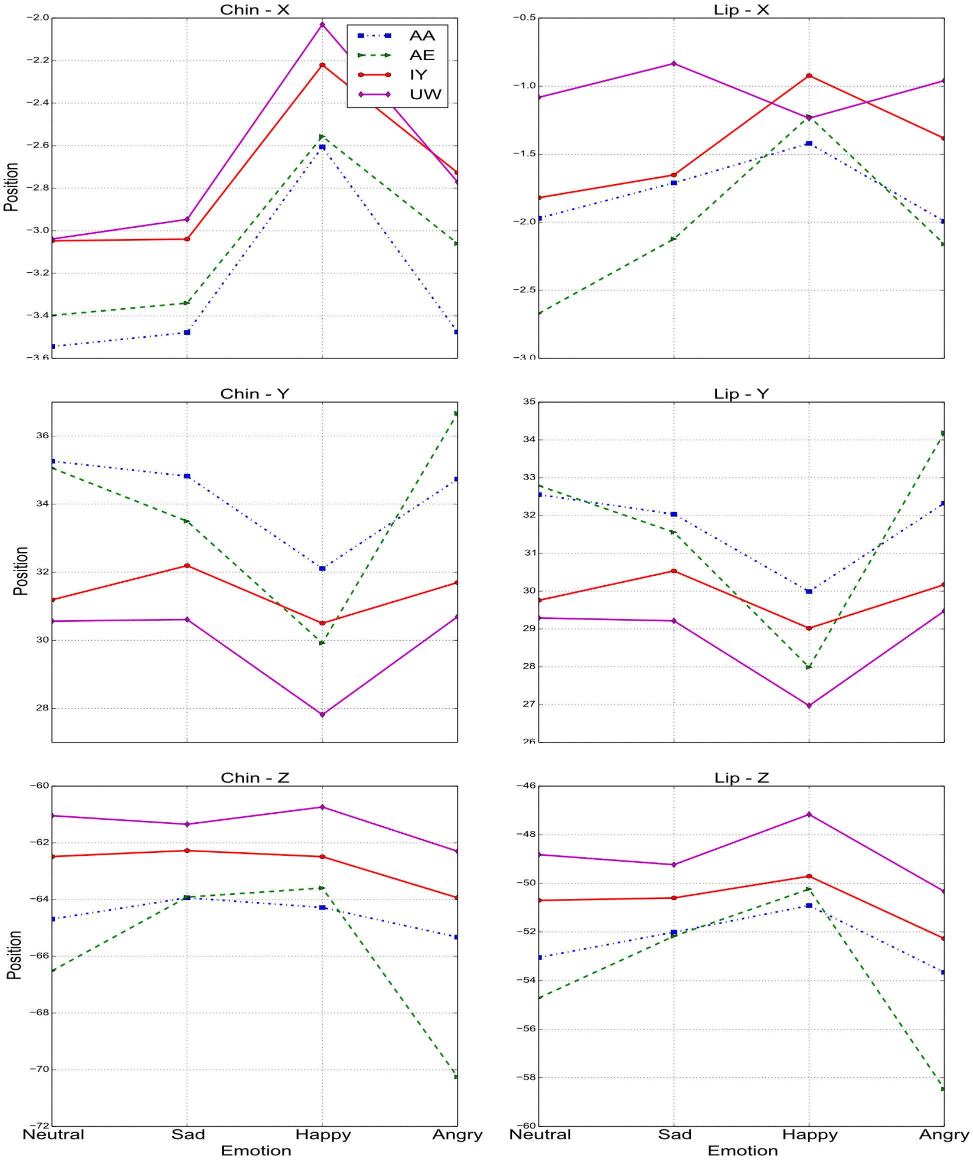


Figure 4.4: Mean Position of Select Articulatory Targets across Different Vowels and Emotions in USC IEMOCAP. (Top) X, (Middle) Y, and (Bottom) Z Coordinates.

Table 4.1: Articulatory Targets for USC EMA and USC IEMOCAP.

Location	Attributes	Axes	Total
USC EMA			
Tongue (TNG)	Position (POS)	(X, Y)	18
Jaw (JAW)	Velocity (VEL)		
Lip (LIP)	Acceleration (ACC)		
USC IEMOCAP			
Chin (CHN)	Chin Width (CHW)	(X, Y, Z)	15
	Chin Position (CHP)		
Lip (LIP)	Lip Width (LPW)		
	Lip Height (LPH)		
	Lip Position (LPP)		

The position is distinctly different for sadness compared to happiness or anger. The effectiveness of these markers towards emotion recognition was studied in [18].

4.2 Proposed Approach

In this section, the preprocessing and feature extraction routines are first presented, followed by a detailed description of the proposed ACL method.

4.2.1 Preprocessing

The focus of this work is on studying peripheral vowels, */AA/*, */AE/*, */IY/*, and */UW/*. Hence, vowel duration is appropriately chosen as the unit of analysis. The utterances are forced-aligned to obtain the vowel boundaries. For USC EMA, SailAlign [106] tool was used to perform this task. Forced alignment was not necessary for USC IEMOCAP as boundary information was already provided with the database.

Acoustic, low-level descriptors (LLD) such as energy of 26 MFBs, pitch, first two formants and overall intensity (30 LLDs) are extracted using sliding and overlapping windows over the vowel segments with a frame rate of 200 frames/second. Energy and MFBs were extracted using the openEAR toolkit [12], while pitch and formants were calculated using the Praat software [107]. Five statistics including the mean, standard deviation, minimum, maximum and range are from each LLD trajectory. This process results in a 150-dimensional feature vector for each vowel segment. The supra-segmental, acoustic features for the i^{th} vowel segment is denoted by x_i.

A supra-segmental representation, similar to the acoustic features, must be extracted from the articulatory targets. Different statistics can be used for this purpose; the mean value of each articulatory target calculated over the vowel segment is used in this study. The k^{th} articulatory target of the i^{th} vowel segment is denoted by a_i^k.

Each vowel segment is assigned the same emotion that external evaluators assigned to the complete utterance. In case of binary classification, the four emotions are split in two categories depending on the task under consideration - Arousal (happy/angry vs. neutral/sad), and Valence (happy/neutral vs. angry/sad). The emotion label for the i^{th} vowel segment is denoted by y_i.

4.2.2 L1-Regularized Logistic Regression

A traditional, cross-entropy error based cost function for logistic regression over the acoustic features x and binary emotion labels y of N segments is defined by Eq (4.1).

$$f(w) = -\frac{1}{N}[\sum_{i=1}^{N} y_i \log \sigma(w^T x_i) + (1 - y_i) \log(1 - \sigma(w^T x_i))] \tag{4.1}$$

The logistic or sigmoid function, $\sigma(\cdot)$, is given by Eq (4.2).

$$\sigma(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}} \tag{4.2}$$

Training involves the learning of optimal weights $w^\star$, by minimizing the given cost function. Recognition involves calculating the posterior probability and label assignment as per Eq (4.3).

$$p(y = 1|x; w) = \sigma(w^T x) \tag{4.3}$$

To prevent overfitting and learn a sparse weight vector, it is common practice to modify the cost function to include an additional L1-regularization term, as given in Eq (4.4).

$$f_{L1}(w, \lambda_1) = f(w) + \lambda_1 \|w\|_1 \tag{4.4}$$

This regularization has the added benefit of making this method suitable for training sets with fewer samples and relatively larger (high-dimensional) feature sets [108]. The above problem is convex and a number of fast techniques have been proposed in literature [109, 110, 111].

4.2.3 Articulation Constrained Learning

In order to improve emotion recognition performance using acoustic and articulatory information, the proposed articulation constrained learning method is devised by further modifying Eq (4.4) to Eq (4.5).

$$f_{ACL}(w, \lambda_1, \lambda_2) = f_{L1}(w, \lambda_1) + \frac{\lambda_2}{M} \sum_{j=1}^{M} (a_j - w^T x_j)^2 \tag{4.5}$$

Here, an additional regularization term is included to minimize the mean squared error over articulatory target reconstruction. λ_1 controls the sparsity of the solution,

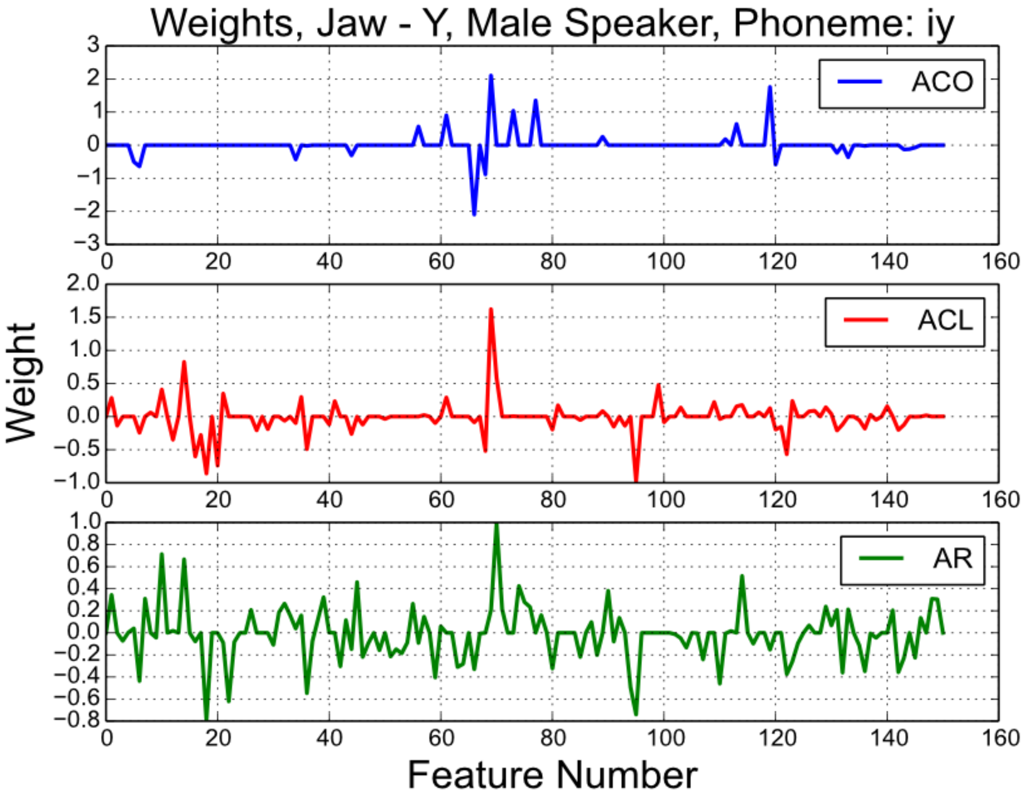


Figure 4.5: A Comparison of Weight Vectors Learnt from Different Training Criteria. (Top) ACO: Emotion Recognition Using Only Logistic Regression, (Middle) Proposed ACL Method, and (Bottom) AR: Least Squares Regression over the Articulatory Target Only.

while λ_2 controls the importance given to articulatory target reconstruction relative to classification. Thus, the optimal weight vector is learnt by jointly optimizing over two tasks: (i) articulatory target reconstruction, and (ii) emotion recognition. The hypothesis is that the correlation between the two tasks is expected to reflect in the weights and lead to an improvement in classification accuracy. Additionally, the proposed ACL cost function has the following important properties:

Firstly, note that the term related to articulatory target reconstruction is not required to operate on the acoustic features belonging to the same database as used for emotion classification. This allows for the flexibility to jointly optimize over features corresponding to different speakers belonging to the same or different database. Consequently, ACL is applicable in scenarios where limited articulatory data is available, but, acoustic data is available in abundance. Secondly, in spite of the additional regularization term, the posterior probability calculation remains the same as Eq (4.3). Hence, articulatory data is not required during the recognition step, which is an appealing property, owing to the difficult and time-consuming procedures for articulatory data collection. Lastly, the L1-regularization term enforces the weight vector to be sparse, thus, providing a shared and interpretable representation of features that contribute towards both tasks.

A particular example highlighting the last aspect is shown in Figure 4.5 and Table 4.2. Figure 4.5 shows a comparison of weights learnt under different objective functions - (i) only emotion recognition or ACO, (ii) ACL, and (iii) least squares regression over articulatory targets or AR. Whereas, Table 4.2 displays the 5 features for each model, ranked on the basis of the magnitude of their weights. One can observe that ACL is able to learn features from multiple tasks, while simultaneously improving the recognition accuracy.

4.2.4 Extension to Multiple Targets

The ACL cost function given in Eq (4.5) is suitable for learning from a single articulatory target. As described in Section 4.1, databases often include articulatory information captured from sensors across multiple locations and axes. The proposed modification to ACL for K targets is given by a single cost function in Eq (4.6).

$$f_{ACL}(w, \lambda_1^k, \lambda_2^k) = \sum_{k=1}^{K} [f_{L1}(w_k, \lambda_1^k) + \frac{\lambda_2^k}{M} \sum_{j=1}^{M} (a_j^k - w_k^T x_j)^2] \tag{4.6}$$

Here, the final weight matrix is defined as $w = [w_1, .., w_k, ...w_K]$. Effectively, the

Table 4.2: Example Comparison of Top Ranked Features for Different Training Objective Functions. Results are for Valence Classification for Male Speaker and Vowel */IY/*. Learning is Constrained Using Jaw Position along the Y-axis, with $\lambda_1 = 0.5$ and $\lambda_2 = 0.1$. UAR: Unweighted Average Recall, CC: Correlation Coefficient, MFB: Mel Filter Bank.

	ACO	ACL	AR
Rank	Top Features		
1	**MFB 18, MIN**	**MFB 18, MIN**	MFB 19, MIN
2	MFB 15, MIN	**MFB 19, MAX**	**MFB 19, MEAN**
3	MFB 16, RNG	**MFB 19, MEAN**	**MFB 19, MAX**
4	MFB 26, MIN	**MFB 15, MEAN**	MFB 11, MEAN
5	MFB 22, MIN	MFB 21, MEAN	**MFB 15, MEAN**
UAR (%)	80.1	83.2	-
CC	-0.42	0.78	0.90

function $f_{ACL,M}(w, \cdot)$ is a summation of $f_{ACL}(w_k, \cdot)$ over K targets. Hence, the learning from each target is considered independently from other targets, i.e. $f_{ACL}(w_k, \cdot)$ is solved independently for each of the K targets. During recognition, the posterior probability estimates from the targets are combined as per Eq (4.7) to yield an average estimate.

$$p(y = 1|x; w) = \frac{1}{K} \sum_{k=1}^{K} \sigma(w_k^T x) \tag{4.7}$$

An important benefit of this strategy is that it allows for one to separately measure or investigate the contribution of each target or a group of targets to the overall classification without requiring additional training.

4.2.5 Extension to Multiple Classes

So far, the emotion labels were assumed to be binary, arousal or valence based attributes. An alternative and intuitive representation for emotions is in terms of discrete or categorical attributes such as happy, sad, angry, etc. Popular strategies, such as one-vs-one or one-vs-rest, are suitable for discriminating between multiple classes of emotions. The latter method is adopted here owing to its lower complexity during training. Accordingly, a one-vs-rest classifier is trained for each emotion, i.e. happy vs not happy, and so on. In this case, ACL yields a set of C posterior probabilities corresponding to each of the C emotion categories. The output label is assigned as per Eq (4.8).

$$\hat{y} = \arg\max_{c} p(y = 1|x; w_c) \tag{4.8}$$

4.2.6 Optimization

Keeping the regularization coefficients λ_1^k and λ_2^k fixed, the cost functions specified in Eqs (4.4), (4.5) and (4.6) are convex in w. Fast solvers described in literature either support logistic regression or linear regression individually, but not jointly [111]. In this work, a generic, off-the-shelf toolbox, CVX [112], was used. Optimization is performed using a splitting conic solver (SCS), which yields slightly less accurate estimates, but are considerably faster.

4.2.7 Choosing Regularization Coefficients

Different methods including Bayesian optimization or cross-validation are available to estimate suitable values for λ_1^k and λ_2^k. Here, an exhaustive search across discrete combinations of λ_1^k and λ_2^k is combined with cross-validation. The search is restricted to $\lambda_1^k \in \{0.1, 0.5, 1.0\}$ and $\lambda_2^k \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$. Further details regarding the cross-validation procedures specific to each database are described in Section 4.3.

4.3 Experimental Results: Within-Corpus

Experiments are conducted to evaluate performance in a within-corpus scenario. Here, the articulatory and acoustic data belong to the same database. The constraints involving articulatory target reconstruction are speaker-independent, while speaker-dependent and speaker-independent emotion classification are considered. A purely acoustic features-based logistic regression, i.e. Eq 4.4, is considered as the baseline. The unweighted average recall (UAR), which is the same as the average of class-wise accuracies, is a reliable metric for unbalanced datasets. Hence, this metric is used to draw comparisons between different methods. Statistical significance is calculated using a difference of proportions test, with a significance level of $\alpha = 0.05$.

4.3.1 EMA (Binary Classification)

Experiments are conducted using a Leave-One-Speaker-Out (LOSO) strategy for the 3 speakers in this database. Speaker-dependent acoustic features are used for training logistic regression; the acoustic features used for the articulatory reconstruction part are speaker-independent. Furthermore, a separate model is learnt for each speaker and vowel. A random subset of all the utterances belonging to the test speaker is used for training logistic regression, while the remaining utterances belonging to the same speaker are used for evaluation. The train/test ratio is kept fixed at 0.5 for all speakers and vowels.

Due to the limited amount of data, cross-validation for choosing regularization coefficients λ_1^k and λ_2^k is performed as follows: The process outlined above is per-

Table 4.3: Within-Corpus Results for Binary Classification on USC EMA. The UAR is Expressed in %.

Vowel	ACO	ACL	Best Target	Best Group
			Arousal	
/AA/	95.12	93.46	93.84 (JAW, VEL, Y)	93.51 (JAW)
/AE/	91.47	91.88	91.74 (LIP, POS, X)	92.01 (LIP)
/IY/	95.59	95.74	95.57 (JAW, POS, X)	95.80 (JAW)
/UW/	93.64	93.85	93.64 (TNG, ACC, X)	94.41 (LIP)
			Valence	
/AA/	77.89	79.46	79.38 (JAW, ACC, Y)	79.34 (LIP)
/AE/	78.40	78.34	78.26 (TNG, VEL, X)	78.20 (JAW)
/IY/	69.45	73.04	73.90 (LIP, VEL, Y)	73.01 (LIP)
/UW/	72.01	76.13	77.63 (JAW, ACC, X)	76.30 (LIP)

formed 15 times with different train/test partitions. A discrete combinatorial search is performed to find the coefficients that achieve the best average performance over these runs. The training process is then repeated an additional 10 times over different train/test partitions with the selected coefficients only. The recall performance achieved over the test set of each of the 10 runs is averaged. The entire process is repeated for each vowel and speaker in the database. The final results are presented separately for each vowel, but aggregated across all speakers.

The UA recall for binary, arousal and valence classification tasks are presented in Table 4.3. The average recall is generally quite high for arousal classification. For this task, the proposed ACL method does not yield any significant improvements, partly due to this saturation in performance. On the other hand, for valence classification, the importance of ACL can be clearly observed. For all vowels except */AE/*, there is an improvement in recall performance. Especially for vowels */IY/* and */UW/*, this improvement is statistically significant. A relative increase of 5.1% and 5.7% is observed for */IY/* and */UW/*, respectively, over an acoustic-only approach. Previous studies have shown that acoustic features are, in general, better suited for arousal than valence classification. Information from alternative sources, such as facial expressions, is found to be more suitable for the latter task. Thus, articulatory information, which can be regarded to be closely related to facial expressions, probably accounts for its stronger impact on valence discrimination.

The aforementioned results were observed for the case where the posterior estimates from all the articulatory targets are combined. Table 4.3 also shows the best individual target or group of targets (Table 4.1) for each vowel and task. Of all the groups, the jaw or lip is more useful than tongue information. For instance, marginal improvements are observed for arousal and valence classification of */UW/* if only the information pertaining to lip is used for learning.

4.3.2 EMA (Multi-class Classification)

The results for multi-class or categorical emotion recognition are shown in Table 4.4. The overall performance shown by ACL is better compared to an acoustic-only approach for all vowels. Specifically, emotions in */IY/* are relatively hard to recognize across all vowels. The results obtained here are similar to a previous study on the same database [22]. A direct comparison is not feasible as experiments in the latter study were performed on a single subject and the partitioning strategy for training was not clearly specified.

Once again, observing the performance using individual or a group of targets, it can be seen that the jaw and lip sensors are more valuable for categorical classification as well. The recall performance on vowel */IY/* increases from 72.89% to 74.66% if learning is performed using only the velocity of the jaw along the x-axis.

Table 4.4: Within-Corpus Results for Categorical Classification on USC EMA. The UAR is Expressed in %.

Vowel	ACO	ACL	Best Target	Best Group
/AA/	79.97	81.55	81.17 (TNG, ACC, X)	81.75 (JAW)
/AE/	78.20	79.02	78.55 (LIP, VEL, X)	79.17 (LIP)
/IY/	71.15	72.89	74.66 (JAW, VEL, X)	73.14 (LIP)
/UW/	75.17	77.50	76.31 (JAW, ACC, X)	77.80 (LIP)

Further inferences regarding the performance can be drawn from the confusion matrices shown in Tables 4.5, 4.6, 4.7 and 4.8. Using articulatory information, the accuracy of recognizing happiness across all vowels increases significantly, i.e. from 63.2% to 70.5%. This improvement comes at the expense of a marginal deterioration in recognizing the remaining emotions. For neutral, anger and sadness, the accuracy using ACL is 80.8%, 73.4% and 81.3%, respectively, compared to 82.4%, 74.3% and 81.8%, respectively, using the baseline approach. It can also be observed that articulatory information serves to reduce the confusion in discriminating emotions across the valence axis, i.e. between happy-angry or neutral-sad. Here, the rate of misclassifying neutral as sadness is 11.7% for ACL compared to 14.5% for the baseline. Similarly, the rate of misclassifying happy as angry is 16.1% for ACL compared to the baseline 20.4%.

Previous studies on multi-modal emotion recognition also showed that facial expressions, especially those captured from the lower portion of the face, i.e. mouth and chin, are better at recognizing happiness compared to other emotions. Once again, the strong relation between articulatory information and facial expressions, probably explain the similar performance results obtained in this work.

Table 4.5: Confusion Matrix for */AA/* in USC EMA. Rows Represent the Ground Truth, While, Columns Indicate the Recognized Emotion. Results are in %.

	N	A	S	H
N	**81.4**	1.9	11.4	5.3
A	8	**75.5**	1.3	15.2
S	13.3	0	**85.5**	1.2
H	3.8	20.2	5.9	**70**

(a) */AA/* (ACO)

	N	A	S	H
N	**89.6**	3.1	4.6	2.7
A	8.8	**72.5**	1.6	17.1
S	12.6	1	**85**	1.4
H	5.2	15.7	7.3	**71.8**

(b) */AA/* (ACL)

Table 4.6: Confusion Matrix for */AE/* in USC EMA. Rows Represent the Ground Truth, While, Columns Indicate the Recognized Emotion. Results are in %.

	N	A	S	H
N	**83.9**	1.6	10.8	3.7
A	7.5	**77.6**	4.6	10.3
S	12.3	0.7	**81.4**	5.6
H	6.7	16.8	9.7	**66.9**

(a) */AE/* (ACO)

	N	A	S	H
N	**84.7**	3.1	10.2	1.9
A	7.7	**75**	3.4	14
S	11.2	0.9	**80.8**	7.2
H	5.2	12.7	10.1	**71.9**

(b) */AE/* (ACL)

4.3.3 USC IEMOCAP (Binary Classification)

Experiments are conducted using a Leave-One-Speaker-Out (LOSO) strategy for the 10 speakers in this database. Speaker-independent acoustic features are used for training both, logistic regression and articulatory reconstruction. Separate models are learnt for each speaker and vowel. The regularization coefficients, λ_1^k and λ_2^k, are chosen on the basis of the best average recall over all speakers. Similar to the experiments carried out on the USC EMA database, separate models are learnt for each speaker and vowel. The final results are presented separately for each vowel, but aggregated across all speakers.

Table 4.7: Confusion Matrix for */IY/* in USC EMA. Rows Represent the Ground Truth, While, Columns Indicate the Recognized Emotion. Results are in %.

	N	A	S	H
N	**77.2**	1.8	18.3	2.8
A	4	**63.1**	4	28.9
S	15.6	0.5	**81**	2.9
H	8.8	27.7	6.7	**56.8**

(a) */IY/* (ACO)

	N	A	S	H
N	**70.8**	1.8	18.3	9.1
A	2.2	**71.6**	3.5	22.7
S	12.4	2.4	**80.7**	4.4
H	5.6	27.4	4.6	**62.5**

(b) */IY/* (ACL)

Table 4.8: Confusion Matrix for */UW/* in USC EMA. Rows Represent the Ground Truth, While, Columns Indicate the Recognized Emotion. Results are in %.

	N	A	S	H
N	**77.2**	2.3	17.3	3.3
A	5.2	**80.8**	6.6	7.4
S	16.4	0.6	**79.5**	3.5
H	13.2	17	10.6	**59.1**

(a) */UW/* (ACO)

	N	A	S	H
N	**77.9**	2.9	13.7	5.5
A	5.9	**74.5**	7.7	11.8
S	15.9	0.9	**78.7**	4.6
H	8.1	8.9	7.2	**75.7**

(b) */UW/* (ACL)

The UA recall for binary, arousal and valence classification tasks are presented in Table 4.9. For the former task, the proposed ACL method performs slightly worse or better depending on the vowel. The best improvement is achieved for the vowel */AA/*, a relative increase of 2.6% over the baseline. For valence classification, ACL outperforms the baseline for each vowel, yet, the results are not statistically significant. Here, the best improvement is achieved for the vowel */AE/*, a relative increase of 2.5% over the baseline.

The results above are for the case where the posterior estimates from all the articulatory targets are combined. Table 4.9 also shows the best individual target or group

Table 4.9: Within-Corpus Results for Binary Classification on USC IEMOCAP. The UAR is Expressed in %.

Vowel	ACO	ACL	Best Target	Best Group
	Arousal			
/AA/	67.43	69.18	72.60 (LPH, Z)	70.28 (LPH)
/AE/	64.24	65.40	66.95 (LPH, X)	65.74 (LPH)
/IY/	66.36	65.85	67.90 (LIP, Y)	67.70 (LIP)
/UW/	63.90	64.44	67.60 (LPH, Z)	66.50 (LIP)
	Valence			
/AA/	60.47	60.68	64.69 (LIP, Z)	63.01 (CHW)
/AE/	57.61	59.06	61.49 (LPW, X)	60.02 (CHW)
/IY/	60.36	61.34	62.85 (CHN, Y)	61.85 (LPW)
/UW/	63.17	63.62	64.42 (CHW, X)	63.95 (LIP)

of targets (Table 4.1) for each vowel and task. Among the different groups of targets or markers, the lip markers are more useful than the chin markers. Articulation constraints using only a single target yields a better performance in comparison to learning from all available targets. For instance, using only the lip height data along the z-axis, statistically significant improvements are obtained for arousal and valence classification over */AA/*. The relative improvement over the baseline is 7.6% and 6.9%, respectively. Similarly, a recall of 61.49% is obtained for valence classification over */AE/*, a relative improvement of 6.7% over the baseline.

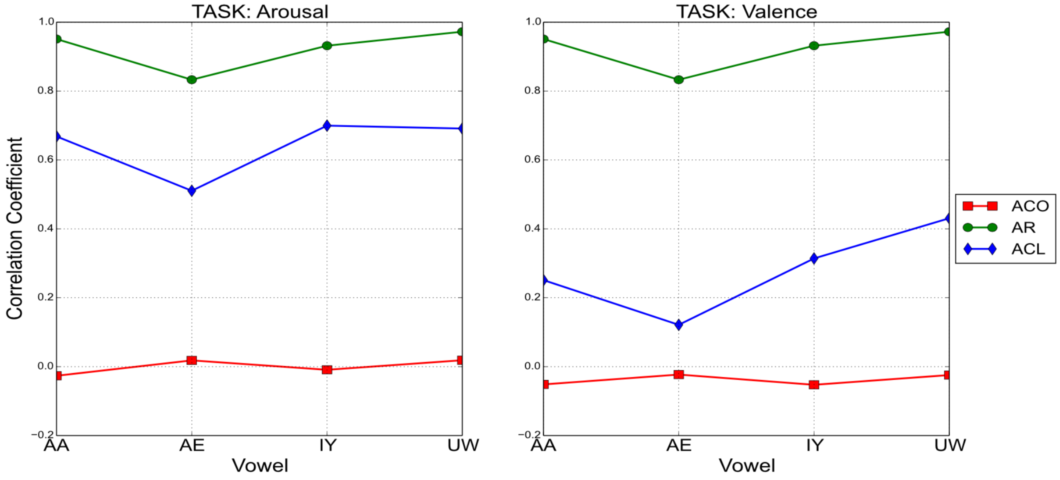
Once again, the recall performance for arousal classification is higher than its valence counterpart. However, the performance is lower relative to USC EMA database. This can mainly be attributed to emotions being more naturally expressed in USC IEMOCAP. The inter-evaluator agreement for this database, measured using the kappa statistic is 0.4 [97]. This low value suggests the difficulty evaluators experienced in labeling these utterances. Secondly, the logistic regression term for emotion classification is completely speaker-independent, which is also known to affect the performance.

4.3.4 Discussion

Experiments over multiple databases show that the proposed ACL method is indeed effective towards improving emotion recognition performance. The impact on arousal classification is lesser compared to valence classification. Performing the latter using only acoustic features is known to be quite difficult; hence, the results obtained in this work are of importance. The expectation here is that via ACL, a shared representation can be learnt that would not only lead to reliable emotion recognition, but also be able to reconstruct articulatory targets based on the constraints.

This property is verified here and the results are shown in Figures 4.6 and 4.7. The correlation coefficient is calculated between the ground truth and reconstructed articulatory targets over all the utterances used during training. This coefficient is calculated for each articulatory target and speaker and the averaged results for each phoneme are presented. The reconstructions obtained via three models are compared: (i) ACO, or Eq (4.4), (ii) ACL or Eq (4.6), and (iii) AR, which is a least squares regression over articulatory targets.

As expected, the AR method shows the maximum CC for all vowels, since it is optimized for a single task; i.e. articulatory target reconstruction. At the opposite end, the ACO method is bound to perform the worst as articulatory data is not considered at all. The ACL method, which learns both tasks simultaneously, shows a higher CC than ACO. For the same weights, the emotion recognition performance is also better than a purely acoustics driven methods such as ACO. From Figure 4.6 and 4.7, a few differences between USC EMA and USC IEMOCAP can also be observed.



(a)

(b)

Figure 4.6: A Comparison of the Average Correlation Coefficient over All Articulatory Targets and Speakers under Different Training Criteria for USC EMA.

The overall CC on the former is relatively higher compared to the latter. This can probably be attributed to the manner in which articulatory data was collected. For USC EMA, the sensors are directly attached to the places of articulation such as lip or jaw. On the contrary, for USC IEMOCAP, this data was collected in the form of motion capture markers, which were originally intended for facial expressions analysis.

4.4 Experimental Results: Cross-Corpus

Here, cross-corpus is with reference to the sources of data for emotion recognition and articulatory reconstruction, i.e. the acoustic data used for articulatory target reconstruction belongs to a different corpus from the one used for training the logistic regression classifier. Experiments are conducted using the same databases, USC EMA and USC IEMOCAP. Recognition performance is evaluated in terms of the UA recall and the within-corpus results obtained in Section 4.3 serve as the baseline.

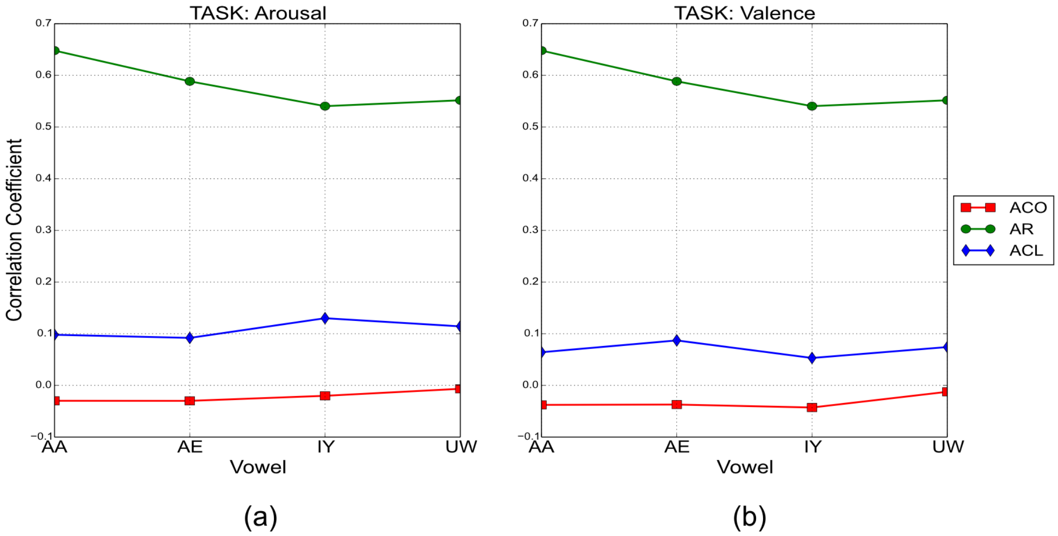


Figure 4.7: A Comparison of the Average Correlation Coefficient over All Articulatory Targets and Speakers under Different Training Criteria for USC IEMOCAP.

4.4.1 USC EMA

In the first experiment, USC EMA is designated as the test corpus, hence, acoustic features from this database are used to train the logistic regression function in a speaker-dependent manner. Acoustic features and articulatory targets, available from 15 facial markers (Table 4.1) and the 10 speakers of USC IEMOCAP, are used for the constraints forcing articulatory reconstruction. The training and cross-validation procedure is the same as the one outlined in Section 4.3.

The recall performance for arousal and valence classification tasks is presented in Table 4.10. Overall, there is no significant impact on recall performance in spite of the data being collected from different sources. The deterioration relative to the baseline, within-corpus performance is severe only for select cases: arousal classification of vowel */IY/* and valence classification of vowel */UW/*. The relative drop in performance is 5% in the former case, and 3.4% in the latter case. In a few case, an

Table 4.10: Cross-Corpus, Binary Classification, Test on USC EMA.

Vowel	ACL (WC)	ACL (CC)	Best Group
	Arousal		
AA	93.46	93.85	93.99 (LPP)
AE	91.88	89.59	90.39 (LPH)
IY	95.74	90.89	89.90 (LPH)
UW	93.85	91.04	90.40 (LPH)
	Valence		
AA	79.46	79.65	80.32 (LPW)
AE	78.34	79.50	79.62 (LPW)
IY	73.04	73.16	73.42 (CHW)
UW	76.13	73.51	74.39 (LPH)

improvement in performance is also observed; for instance, valence classification over vowel */AE/*.

4.4.2 USC IEMOCAP

In the second experiment, USC IEMOCAP is designated as the test corpus. Here, acoustic features from this database are used to train the logistic regression function in a speaker-independent manner. Acoustic features and articulatory targets from all the 18 targets (Table 4.1) and 3 speakers of USC EMA are used for the term involving articulatory reconstruction constraints. The training and cross-validation procedure is similar to the one outlined in Section 4.3.

The recall performance for arousal and valence classification is presented in Table 4.11. The impact of cross-corpus training is minimal on recall performance. Similar to USC EMA, there is an improvement in recall for select cases: arousal classification

Table 4.11: Cross-Corpus, Binary Classification, Test on USC IEMOCAP.

Vowel	ACL (WC)	ACL (CC)	Best Group
		Arousal	
AA	69.18	68.71	70.11 (TNG)
AE	65.40	65.01	66.02 (TNG)
IY	65.85	66.96	67.59 (JAW)
UW	64.44	63.55	65.57 (TNG)
		Valence	
AA	60.68	62.41	62.80 (TNG)
AE	59.06	60.83	61.92 (TNG)
IY	61.34	61.27	61.28 (JAW)
UW	63.62	63.02	63.23 (JAW)

for vowel $/IY/$ and valence classification for vowels $/AA/$ and $/AE/$.

The results above show that cross-corpus training is not severely detrimental to the performance. Marginal deteriorations are expected in certain cases as there are notable differences across corpora [100, 113, 19, 101]. These differences exist among the selection of speakers, recording conditions and types of emotional expressions. Considering these aspects, the cross-corpus performance obtained using the proposed ACL approach is quite promising and allows for better generalization and flexibility to large scale, real-world studies.

4.5 Summary

An articulation constrained learning method was proposed to perform emotion recognition using both acoustic and articulatory information. A conventional L1-regularized logistic regression cost function was extended to jointly optimize two tasks

- (i) emotion classification via logistic regression, and (ii) articulatory reconstruction via least squares regression. The proposed method was extended to consider constraints from multiple articulatory targets as well as categorical emotion recognition. A strong advantage offered by ACL is the inherent flexibility to combine data from different domains without requiring large-scale articulatory data collection.

Experiments were performed to evaluate speaker dependent as well as independent emotion recognition performance on two databases, USC EMA and USC IEMOCAP, providing articulatory information in different manners. On USC EMA, significant improvements of 5.1% and 5.7% were obtained for valence classification of vowels */IY/* and */UW/*, respectively. In comparison, on USC IEMOCAP, an improvement of 2.5% was obtained for the same task on vowel */AE/*. Discriminating across the valence axis is quite challenging using speech, hence, the results obtained in this work demonstrate the importance of articulatory information towards improving the performance on this task. The performance using individual targets was also presented. In this case, an improvement of 6.9% and 6.7% was obtained for valence discrimination of vowels */AA/* and */AE/*, respectively, on USC IEMOCAP. These results show that domain knowledge can be incorporated to improve the performance, i.e. if the relationship between articulatory target behaviors and vowels is known beforehand, then other targets can be given relatively lower importance during the decision-making process.

For categorical emotion recognition on USC EMA, ACL was found to improve the overall performance across all four vowels. Incorporating articulatory constraints was shown to significantly improve the rate of recognizing happy emotions; an 11.55% improvement relative to the baseline was observed. An analysis of the confusion matrices showed that ACL tends to decrease the misclassification rate between emotions with similar arousal characteristics, i.e. happy-angry or neutral-sad. This observation is also supported by the improvement in valence discrimination described above.

Cross-corpus studies were conducted to evaluate generalization ability across different recording conditions, speakers and expression types. Articulatory data available from one database was used to constrain emotion classification over acoustic features belonging to another database. The performance in this scenario was observed to be almost similar to the within-corpus scenario, except for select cases. The deterioration observed in these cases is a commonly expected behavior inherent to cross-corpus studies.

Chapter 5

ARCHITECTURES FOR SPOKEN KEYWORD DETECTION

A keyword detection system acting as the front-end for a speech recognition engine needs to be *always on*, i.e. continuously listening. As a result, there is a strong need to develop an architectural framework for keyword detection with minimal power consumption.

There is a vast amount of literature identifying various methods for keyword detection. Existing methods can be broadly classified as follows - (i) perform complete speech recognition over the phrase and then detect the keyword by looking at the transcriptions provided [23, 24, 25], (ii) train separate models for the keyword and out-of-vocabulary (OOV) words, and detect keywords based on the likelihood over each model. The first method requires the entire phrase to be uttered completely, i.e. offline. It also requires a complete ASR system, which is computationally intensive because of the exhaustive search required to perform transcription. The second method is relatively simple and can be performed in an online setting. It is more suited for applications where the set of keywords to be detected is known beforehand.

Until recently, techniques based on GMMs for acoustic modeling and HMMs for modeling the sequence of words were quite common [26, 27, 28, 29, 30]. The OOV words were modeled using a garbage or a filler model, while a separate GMM-HMM was trained for each keyword. The most likely state sequence was then identified using the Viterbi algorithm. GMMs can be easily implemented in a parallel fashion, however, the Viterbi step is inherently sequential, which increases the computational latency.

Recently, neural network (NN) based methods have shown tremendous success on speech recognition tasks. This success has come after advances made in the field of deep learning, which allows for efficient training of a network with many hidden layers and a large number of neurons (nodes) per layer [31, 32]. These networks are well-suited to capture the complex, non-linear patterns from the acoustic properties of speech. Detection is again straightforward; a matrix-vector multiplication step followed by a non-linear operation at each layer. Such operations can be easily extended for parallel implementations, thus offering a lower latency and a uniform architecture compared to the aforementioned HMM-based methods. One such approach for keyword detection was presented in [32]. In spite of the low-latency algorithm and highly accurate detection performance, the network is quite large, requiring upto a few million multiplications every few milliseconds as well as large memory banks for storing these weights. Mobile devices are often constrained in the amount of available hardware resources, making this approach less suited for practical applications. In this chapter, a NN-based architecture is presented for keyword detection. Special emphasis is placed on reducing the memory and computational overhead using different techniques.

5.1 Proposed Approach

5.1.1 Preprocessing

The Resource Management (RM) database [114] consists of phrases recorded for scenarios pertaining to the naval forces. Speech is processed at a frame rate of 100 frames/second, i.e. a window size of 25ms and step size of 10ms. The first 13 MFCCs are extracted for each frame. These features are augmented with MFCCs of the 15 previous frames and 15 future frames to form a 403-D feature vector per frame.

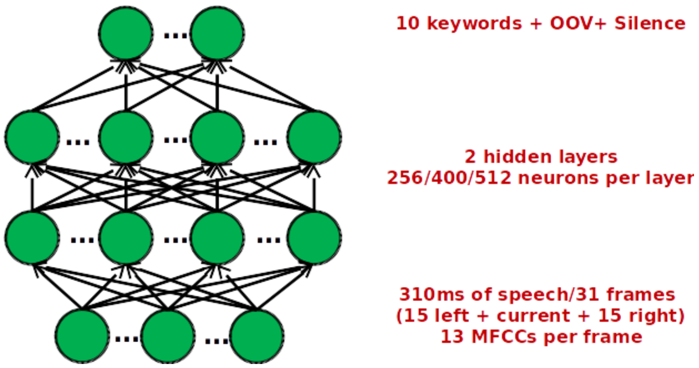

10 keywords + OOV+ Silence
2 hidden layers
256/400/512 neurons per layer
310ms of speech/31 frames
(15 left + current + 15 right)
13 MFCCs per frame

Figure 5.1: A Neural Network Architecture for Keyword Detection.

This corresponds to 31 frames of 310ms of speech; the average word duration for this database was 300ms, hence, this choice was deemed to be appropriate for modeling words or sub-word units. Ten keywords - *ships*, *list*, *chart*, *display*, *fuel*, *show*, *track*, *submarine*, *latitude* and *longitude* were selected in this work. Forced-alignment is performed using the Kaldi speech recognition toolkit [115] in order to obtain the word boundaries. Each frame is labeled as either one of the 10 keywords or OOV or silence. The speaker-independent train and test partitions are already specified with the database; there are 109 and 59 speakers in the training and test set, respectively. The speech features are z-normalized to zero mean and unit variance for each speaker.

5.1.2 Neural Network

The feedforward neural network is shown in Figure 5.1. The network consists of an input layer, two hidden layers and an output layer. The input layer consists of 403 nodes corresponding to the MFCCs extracted above. Denoting the input layer as x_i, where $i = 1, 2, ..., N$ is the number of nodes in the input layer, the computations involved for the input layer to the first hidden layer (h^1) are given as -

$$z_j^1 = \sum_{i=1}^{N} W_{ij}^1 x_i + b_j^1 \tag{5.1}$$

Here W^1 and b^1 refer to the weights and biases of this layer. A non-linear, rectified linear operation [116] is then applied over these intermediate values. Rectified linear (ReLU) units have attained popularity as opposed to the conventional sigmoid/logistic function as they capture more detailed information. Furthermore, they are relatively straightforward to implement in hardware as they require only a comparison operation, according to Eq (5.2). In comparison, a sigmoid operation is typically implemented using a Taylor series expansion and is costly.

$$h_j^1 = \max(0, z_j^1) \tag{5.2}$$

The computations from the first hidden layer to the second hidden layer are the same as Eqs (5.1) and (5.2). The output layer is modeled as a softmax layer with $K + 2$ nodes. Out of these, K nodes correspond to the K pre-defined keywords that are to be detected and the remaining 2 nodes correspond to OOV and silence. The softmax output yields a probability estimate for each of the K possible outputs for the current frame.

Training is performed by minimizing the cross-entropy error cost function. Backpropagation is applied to iteratively update the weights and biases of each layer. Mini-batch stochastic gradient with a batchsize of 500 samples is used for optimization. The network is trained for a total of 10 epochs with a learning rate of 0.001 and a momentum of 0.8. The number of layers vary from 1 to 3, while the number of nodes for each hidden layer range from 256 to 512. The optimal values were determined via validation on a randomly selected subset of the training set.

5.1.3 Post-Processing

The output layer returns a posterior probability estimate for each frame, i.e. every 10ms. To reduce the inherent noise in such estimates, the latter are smoothed using a symmetrical moving average window of W frames centered around the current frame. This helps eliminate noisy bursts and reduce the false alarm rate. The window size is chosen from $W \in \{23, 27, 31, 35, 39\}$. The best window size was found to be $W = 31$ in our experiments. The overall goal is to determine whether a specific keyword is present in the entire phrase, hence, the output should either be 1, if the keyword is present, and 0 otherwise. To obtain this phrase-level decision, an additional post-processing step is applied over the smoothed estimates. Using a sliding window of size C frames, if the average probability estimate within this window exceeds a certain threshold, then a keyword is said to be present in the phrase. This window size C is dependent on the length of the keyword and is chosen from $C \in \{35, 39, 43, 47, 51\}$. The best window size was found to be $C = 51$ in our experiments.

5.1.4 Fixed-Point Implementation

The aforementioned training procedure is implemented using a floating-point representation. The optimized weights, when stored in floating point require a lot of memory. For instance, storing each weight in 32-bit floating-point format would require 2 MBs for a network with 512 nodes per hidden layer. Often, hardware on mobile devices is constrained in the amount of memory available, such as a few KBs only. Hence, a fixed point implementation is necessary to reduce the memory footprint. A histogram of the weights for each layer is shown in Figure 5.2. The weights are normally distributed, and so we can use different linear or non-linear quantization schemes. We follow a simple linear quantization scheme owing to its simplicity

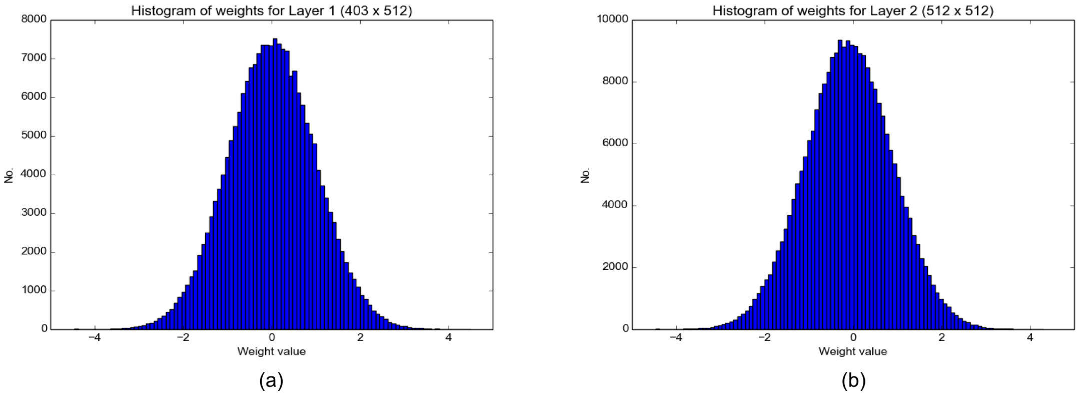


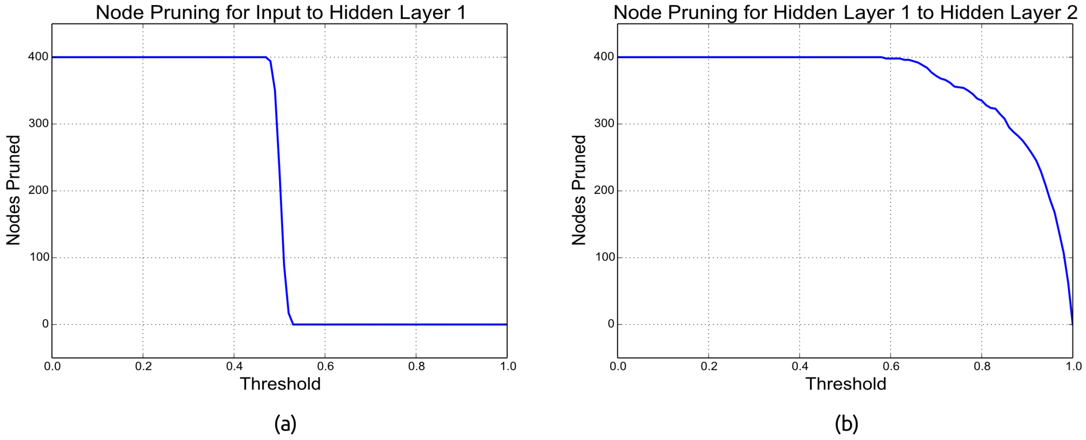
Figure 5.2: Histogram of Weights for (a) Input to Hidden Layer 1, and (b) Hidden Layer 1 to Hidden Layer 2.

and generalizability. Throughout the paper, we denote fixed-point using a Q$A.B$ format, where A denotes the number of bits assigned to the integer part and B denotes the number of bits assigned to the fractional part. Unless mentioned otherwise, an additional sign bit is assumed.

The input nodes and intermediate hidden layers are also stored in a fixed-point format to further reduce the accumulator size during multiplication operations. The former are represented using 16 bits in a Q2.13 format. The latter are represented using 24 or 32 bits, i.e. a Q8.16 or Q16.16 format. The hidden layer nodes are always positive, hence, a sign bit is not required.

5.1.5 Node Pruning

Depending on the size of the neural network, there may be a few nodes in the hidden layers that are rarely or never active. If such nodes can be identified, then they can be pruned away, thus reducing both memory and multiplications. Here, we propose one such approach to identify inactive nodes, which is described below. First, we evaluate the network on the training data and the weights learnt from



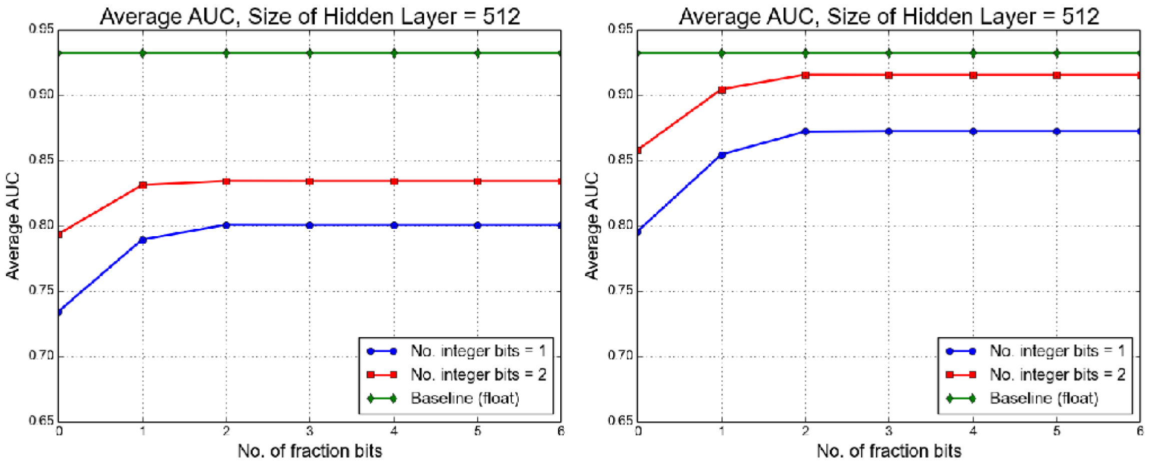
(a)

(b)

Figure 5.3: Number of Nodes Pruned for Different Threshold Values (a) First Hidden Layer, and (b) Second Hidden Layer.

backpropagation during training. For each node in the hidden layers, we identify the nodes which are zero and maintain a count. This count is averaged over all the training examples to yield a probability estimate for each node, i.e. p(node is zero). Using a threshold value $t \in (0, 1)$, we remove the nodes that have $p > t$. The number of nodes pruned for different threshold values t, and for both hidden layers is shown in Figure 5.3. Here, we can observe that for the first hidden layer, there is a sharp change in the number of nodes pruned at $t = 0.5$. For $t < 0.5$, all nodes are pruned away, while for $t > 0.5$, all nodes are retained. In this case, all nodes in the first hidden layer are equally informative and node pruning is not helpful. On the other hand, for the second hidden layer, we can see that the transition is smoother, especially for $0.7 < t < 1$. If we set the threshold in this range, we can expect to prune nodes for only a marginal loss in performance.

Besides node pruning, singular value decomposition (SVD) was also considered for reducing the memory footprint as described in [117]. Accordingly, the weight matrix is represented as a product of two low-rank matrices. This technique helps lowers



Hidden Layers (24 bits)

(a)

Hidden Layers (32 bits)

(b)

Figure 5.4: Effect of Different Fixed-Point Representations for Weights on the Overall AUC Performance. The Input Is Represented Using 16 Bits. Hidden Layer Nodes Are Represented Using (a) 24 Bits, and (b) 32 Bits.

the memory, however, at the cost of increasing the number of multiplications. In our experiments, a significant drop in performance was observed using this technique, possibly owing to the relatively smaller network compared to [117]. The degradation was even higher when SVD is combined with a fixed-point representation.

5.2 Experimental Results

The experiments and results for fixed-point keyword detection using the RM database are described in this section. For the baseline, we consider the performance obtained using a simple floating-point representation for all nodes and weights. The area under the curve (AUC), which calculates the area under the receiver operating characteristics (ROC) curve of true positives vs. false positives, is considered as the metric.

Table 5.1: Comparison of AUC and Memory Requirements Between Floating and Fixed Point Implementations for Networks with Different Hidden Layer Configurations.

Hidden Layer Width	AUC Floating-Point	AUC Fixed-Point	# of Weights	Memory KB
256	0.8520	0.8038	172300	102.7
350	0.8960	0.8428	268462	160.1
400	0.9201	0.9098	326812	194.8
512	0.9321	0.9153	475660	283.6

5.2.1 Floating-Point vs. Fixed-Point

A comparison between floating and fixed-point implementations is shown in Figure 5.4. For fixed-point implementation, the input is represented using 16 bits (Q2.13). Figure 5.4 (a) and (b) show the performance with 24 bits (Q8.16) and 32 bits (Q16.16), respectively, for hidden layer nodes. For the weights stored in a Q$A.B$ format, here, $A \in \{1, 2\}$ and $B \in \{1, 2, 3, 4, 5, 6\}$. First, we can see that the performance is significantly better when using 32 bits for the hidden layer nodes. Secondly, reserving 2 bits for the integer part yields a better AUC compared to just 1 bit. For the fractional part, we observe that increasing the resolution beyond 2 bits does not lead to any significant increase.

A summary of the memory requirements is shown in Table 5.1. The input, hidden layer nodes and weights are stored in Q2.13, Q16.16 and Q2.2 formats, respectively. The best performance is shown for hidden layers with 512 nodes per layer. The memory required in this case is 283.6 KB. For 400 nodes per layer, we can see that there is only a marginal loss in performance; an AUC of 0.9098 compared to a floating point representation of 0.9201, while requiring only 195 KBs of memory. Our results are not directly comparable with the results reported in earlier works [32, 118] since

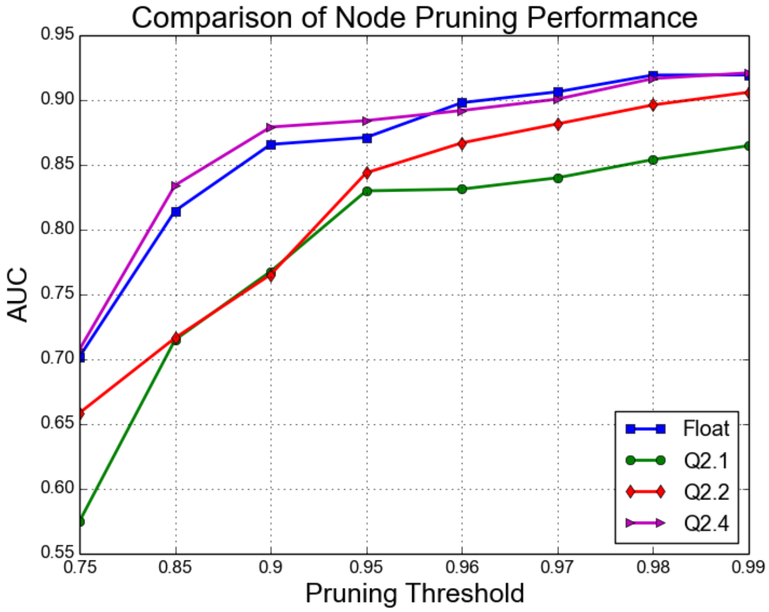


Figure 5.5: A Performance Comparison Between Floating and Fixed-point Implementations for Different Pruning Thresholds.

the databases are completely different. An AUC performance of 0.90 to 0.95 is quite commonly observed for small to medium sized databases. In this aspect, the detection performance obtained here is within an acceptable range.

5.2.2 Node Pruning

The performance after node pruning is shown in Figure 5.5. In this case, only the nodes of the second hidden layer were pruned, as per the procedure described earlier. The performance is analysed for different threshold values $t \in [0.75, 1.0]$. We observe that as the threshold increases, the number of nodes pruned decreases and the performance improves. Furthermore, the figure also shows a comparison between different fixed-point and floating-point representations for the weights. For weights represented in a Q2.2 format, and $t \in [0.95, 1.0]$, the loss in performance is

Table 5.2: Memory Requirements after Pruning for Weights in Q2.2 Format.

Threshold	AUC	# of Weights	Memory (KB)
0.95	0.8438	249581	152.3
0.96	0.8669	257428	157.1
0.97	0.8816	269818	164.6
0.98	0.8962	282621	172.5
0.99	0.9060	301206	183.8
No pruning	0.9098	326812	194.8

not significant. The memory requirements for a network with 400 nodes per hidden layer and a threshold $t \gg 0.7$ are shown in Table 5.2. For $t = 0.99$, the AUC is 0.9060 compared to 0.9098 obtained without pruning. The memory, in this case, reduces by a relative factor of 5%. Similarly, for $t = 0.98$, the AUC is 0.8962 with a relative decrease of 11.4% in memory. Hence, node pruning can be a useful technique to further optimize the neural network and reduce its on-board memory requirements.

5.3 Summary

A fully connected, feedforward neural architecture for spoken keyword detection was proposed in this work. A post-processing method to obtain phrase-level metrics using a sliding window approach was also described. To reduce the memory footprint for network weights, the latter were stored using a fixed-point representation. Experiments were conducted on 10 keywords selected from the RM corpus, and results show that there is only a marginal loss in performance when the weights are stored in a Q2.2 format, i.e. only 5 bits. The total memory required in this case is approximately 200 KBs, making it highly suitable for resource constrained hardware devices. A node pruning technique was also presented to identify and remove the least active nodes in a neural network, thus, decreasing the memory requirements even further. For an acceptable loss in performance, an 11.4% reduction in memory is obtained after combining this technique with a fixed-point representation. These results demonstrate the applicability of the proposed approach for implementations with limited hardware resources.

Chapter 6

LIFELOGGING: INDEXING AND RETRIEVAL OF AMBIENT SOUNDS

The idea of lifelogging is not particularly new and its first appearance in literature was recorded in 1945 [33]. A hypothetical system that would serve as an extension to human memory (Memex) by continuously logging data using wearable devices was described by the author. Recently, *MyLifeBits*, a lifelogging application using a wearable camera, was developed by Microsoft [119]. The underlying algorithm relied heavily on human intervention to annotate different events. Lifelogging specific to ambient sounds was proposed in [35]. A set of tools to achieve automatic segmentation and classification along with a prototype browser for visualization was presented. Although the tools were not fully mature at the time, their research provides good insight into the nature and complexities involved in this problem. Recent advances in the field of audio content analysis can be used to build upon the ideas presented in [35] and make lifelogging with minimal human input a reality. Taking into account the the widespread availability and use of smartphones and tablets, a seamless and intuitive user-interface is strongly desired compared to the interfaces proposed in [119, 35].

A lifelogging application assumes that a wearable device is continuously recording audio for long durations, i.e. upto a few hours in a day. Hence, the most important task here is to devise techniques to break down this long duration recording into discrete events. The goal is to reliably identify when a change of event occurs, commonly known as segmentation. Following this process, an indexing mechanism must exist such that it automatically annotates these individual events with relevant semantic and acoustic tags. This process serves to build an archive of events along with the relevant tags. Lastly, users must be provided with means to search or navigate through

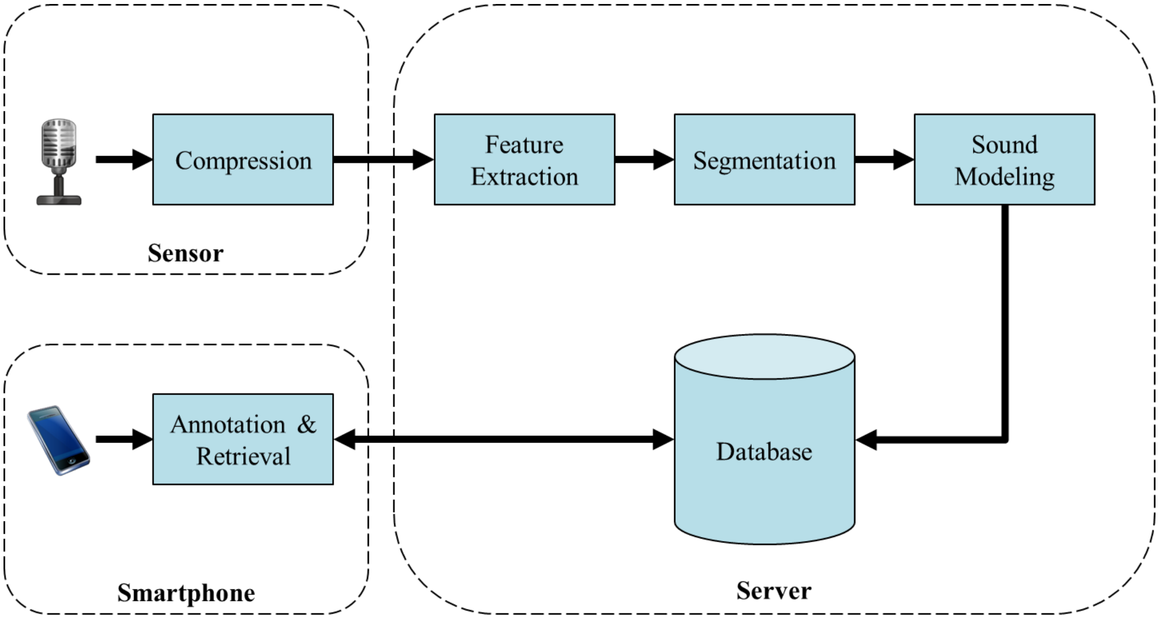


Figure 6.1: Block Diagram of a Typical Lifelogging Application.

their archives and retrieve specific events and information using keywords, tags or examples related to their query.

6.1 Proposed Approach

A complete framework for feature extraction, segmentation, annotation and retrieval of sound events specific to lifelogging is shown in Figure 6.1. Each component in this block diagram is described in detail in the rest of this section.

6.1.1 Feature Extraction

A good set of features must ideally satisfy the following criteria: (i) provide a high-level description of the underlying acoustic content, (ii) work well on different kinds of sound events, including speech and ambient sounds, and (iii) show distinct responses to different events, i.e. encode information suitable for discrimination.

Following these guidelines, a set of spectral and temporal features, as proposed in [36], is used in this work. Features are extracted at different time-scales: (i) short-term features, which include spectral sparsity, spectral centroid and loudness calculated over individual frames of duration 25-50 ms. (ii) long-term features, which include harmonicity, transient index and temporal sparsity calculated over a longer duration of 1-2 s. The features are appropriately synchronized to obtain a 6-dimensional feature vector at every time step. Each clip, T frames long, is described as a feature trajectory, $Y_{1:T}$, where Y_i, $1{\le}i{\le}t$ is a 6-D vector. A detailed analysis on the specific properties of each feature and their extraction procedure can be found in [36].

An example depicting select temporal and spectral features and their responses to different sound events is shown in Figures 6.2 and 6.3, respectively.

6.1.2 Segmentation

There exist a number of approaches to perform segmentation based on event-change detection. Of these, algorithms based on Bayesian Information Criterion (BIC) or a Dynamic Bayesian Network (DBN) have been widely used. In this study, a Switching Linear Dynamical System (SLDS) method is used. This method falls under the general DBN framework. The choice of DBN over BIC is governed by the better performance and lower complexity as demonstrated in [37]. Additionally, this framework is robust; features that are not responsive to certain event changes can be accounted for appropriately. Compared to the BIC approach, DBNs are capable of detecting events of shorter duration, thereby increasing the granularity of segmentation.

The directed acyclic graph (DAG) model employed for segmentation is shown in Figure 6.4. M is the hidden global mode of the frame. It is discrete-valued and can have either of the 3 values - $(ON, OFF, CONT)$. Here, ON indicates that a new

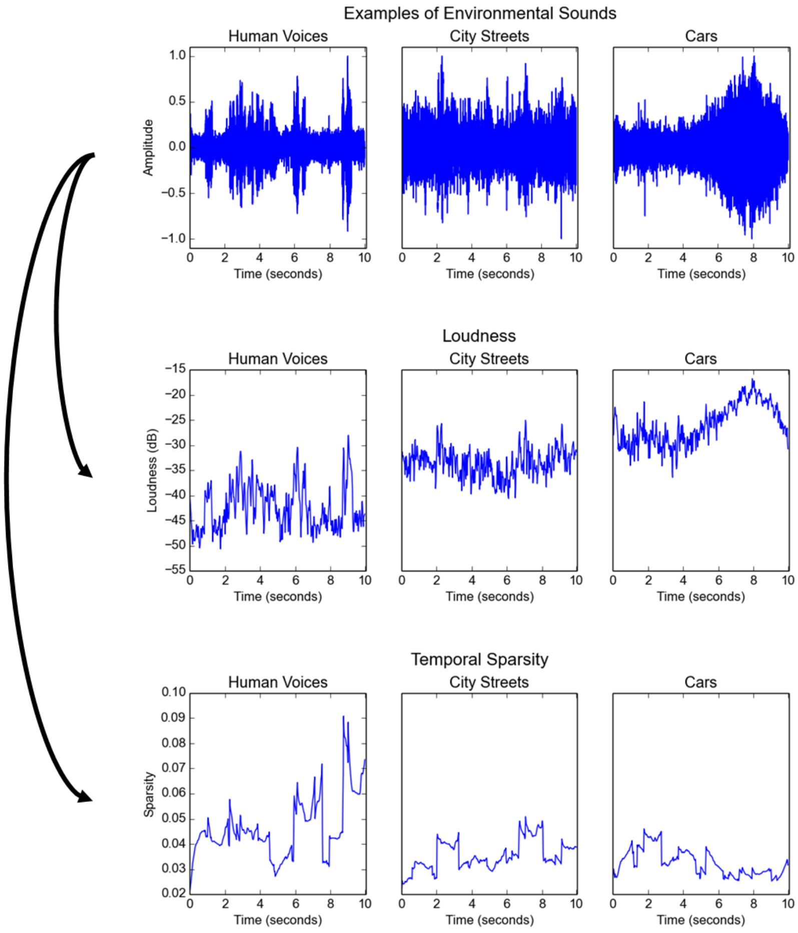


Figure 6.2: A Comparison of Loudness and Temporal Sparsity Features Between 3 Sounds - a Male Speaker, City Noise and Cars.

event has started, *OFF* indicates that the event has ended and *CONT* indicates the continuation of an event from the past frame to the current frame. The hidden nodes, $\mu^{1:K}$, model the responsiveness or delay of each feature (here, $K = 6$) to an event change. μ is discrete-valued and has the same possible values as M. $S^{1:K}$ are continuous-valued, Gaussian, hidden nodes mediating the effect of onsets/end times of individual features on the actual feature observation $Y^{1:K}$.

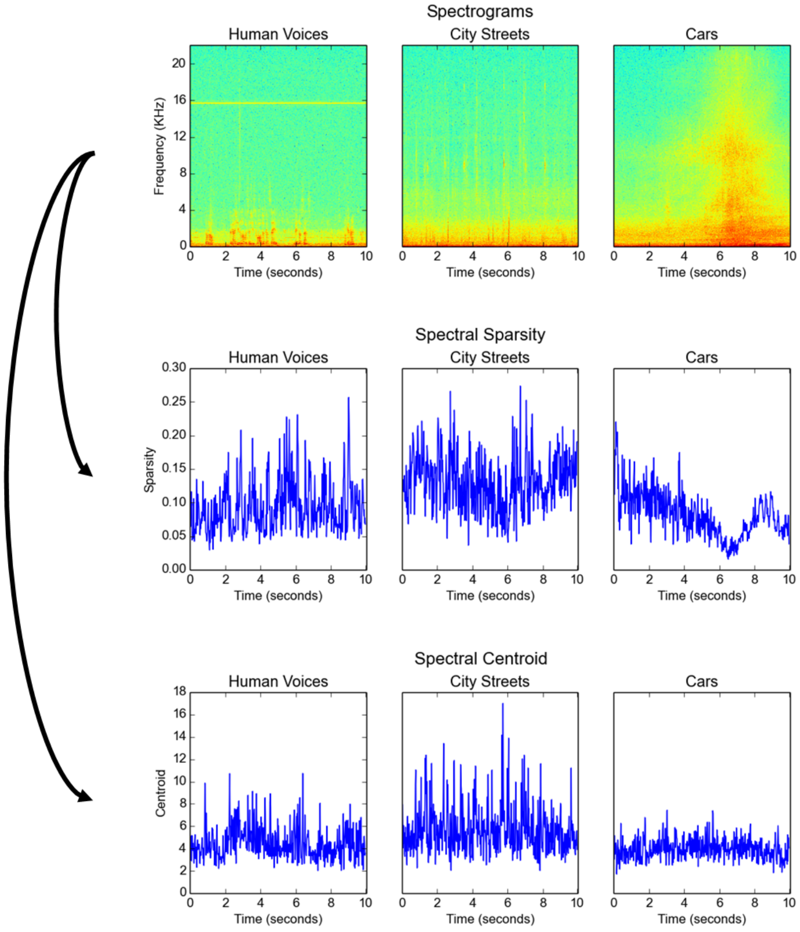


Figure 6.3: A Comparison of Spectral Centroid and Spectral Sparsity Features Between 3 Sounds - a Male Speaker, City Noise and Cars.

$$\hat{M}_{1:T} = \arg\max_{M_{1:T}} P(M_{1:T}|Y_{1:T}^{1:K}) \tag{6.1}$$

The goal of segmentation is then to infer the hidden nodes $M_{1:T}$ from the observed feature trajectory $Y_{1:T}^{1:K}$, using a maximum *a posteriori* (MAP) criterion defined in Eq 6.1. A depiction of the segmentation process is shown via an example in Figure 6.5. Unfortunately, exact inference requires exponential-time complexity. However, a linear-time approximate Viterbi inference scheme exists as described in [37].

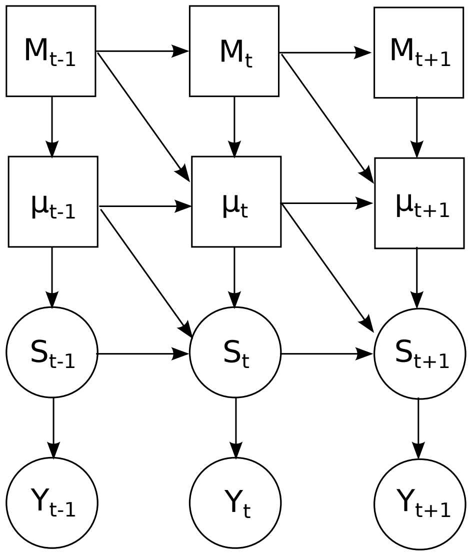
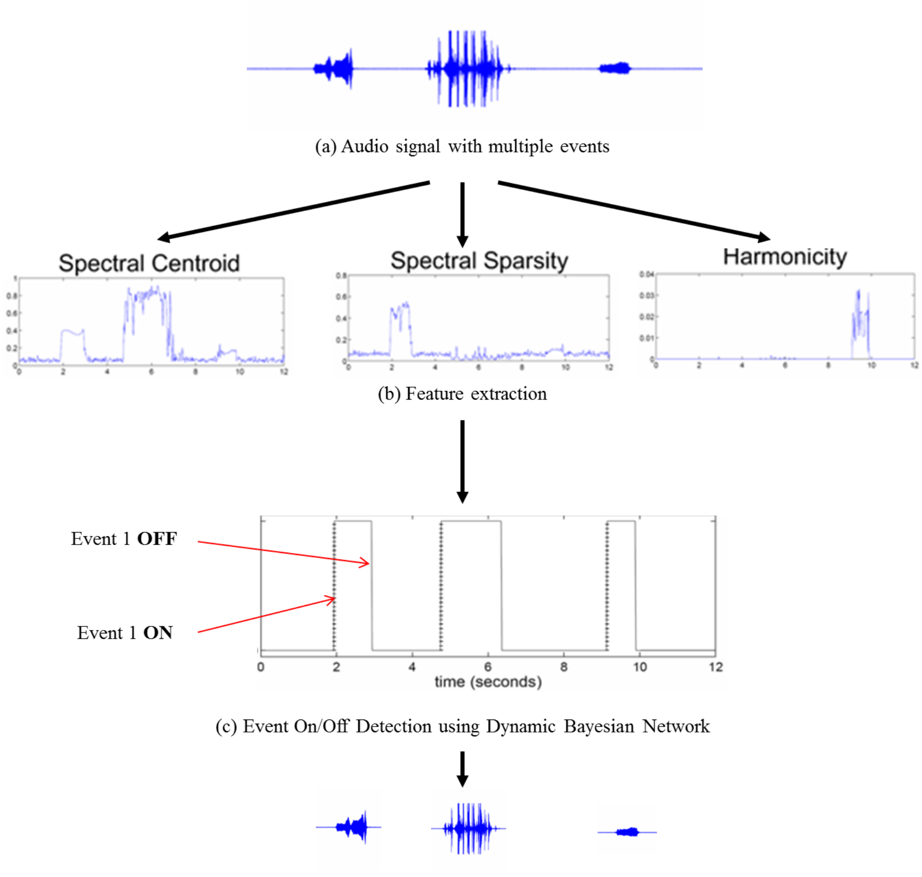


Figure 6.4: SLDS Graph Model for Segmentation. Square and Circular Nodes Represent Discrete and Continuous-Valued Nodes Respectively.

6.1.3 Annotation and Retrieval

Once segmentation returns individual events or clips, they must be automatically annotated with relevant semantic tags and archived to a database for future retrieval. Hence, a technique that facilitates a comparison between different sounds and returns a suitable metric is required. A Query-By-Example (QBE) strategy, described in [37], is adopted here for this purpose. Query behavior is modeled using a likelihood-based strategy; the likelihood is calculated over all possible queries that arise as a result of each sound in the database. To perform annotation, the joint probability $P(X, Y)$ is to be maximized, where query Y is the observation and X is a hidden variable that models the database sound that generated the query. The observed query Y is the feature trajectory of the test sound event. The feature set X also represents a trajectory, however, each individual feature trajectory is a hidden Markov Model (HMM) that approximates the behavior of the trajectory using zero, first and second



(a) Audio signal with multiple events

(b) Feature extraction

(c) Event On/Off Detection using Dynamic Bayesian Network

(d) Segmentation into individual events

Figure 6.5: A Particular Example Describing the Segmentation Procedure.

order polynomial fits. This behavior models the increasing or decreasing nature of the trajectory. Assuming all database sounds are equally likely, i.e $P(X)$ is uniform, all sounds in the database can be ranked with respect to the likelihood $P(Y|X)$. This rank is represented in the form of *sound-sound* weights in the network shown in Figure 6.6.

The *sound-tag* weights are learnt by performing training on a diverse set of sounds and labeling them individually. The higher the association of a tag to a sound by users, the higher the weight between the two nodes. The *tag-tag* weights represent

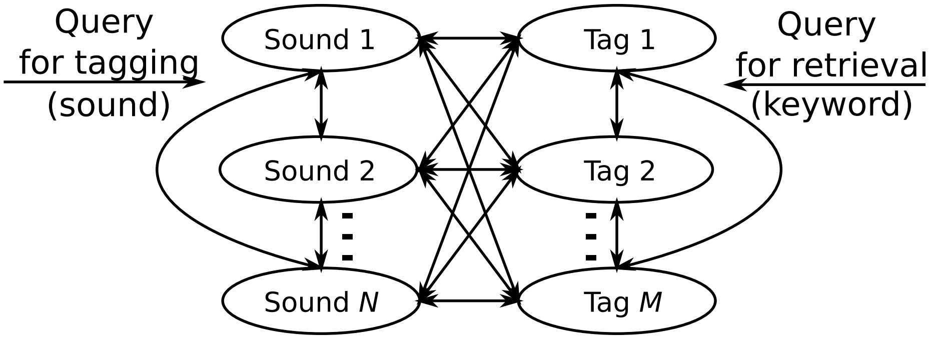


Figure 6.6: A Sound-Sound, Sound-Tag, Tag-Tag Network Used for Annotation and Retrieval.

the semantic similarity between two different tags. To perform automatic annotation, the paths from the query sound to all the tags are evaluated based on the weights between the nodes in Figure 6.6. The path with the highest weight (shortest path) is considered to be the best match and a tag is automatically assigned. In a similar fashion, each sound can be assigned the n highest-ranked tags.

Retrieval is performed in a similar manner, except, the query is now a keyword and the top-ranked sounds associated with that keyword are returned based on the weights between the corresponding nodes in Figure 6.6.

6.2 Experimental Results

All sounds were captured, uncompressed, at a sampling rate of 44100 Hz with 16-bits precision. A training set comprising of diverse sound events and activities was first created. The events are selected in a manner so as to include mundane sounds from conversations, vehicles, workplaces and restaurants as well as rare and interesting sounds like fireworks, alarms and vehicle accidents. A total of 208 sounds selected from the BBC Sound Effects Library, and manually captured using a field recorder, are used for training. A dictionary comprising of 84 different acoustic and

Table 6.1: Performance of QBE-based Annotation.

Tag Rank	Original Feature Set	Enhanced Feature Set
1	64.51%	72.58%
2	59.67%	67.74%
3	54.83%	62.90%

semantic tags such as crowd, talking, machine, explosion, and so on, is created for annotation. Finally, each sound event is labeled with the appropriate tags to build a *sound-sound*, *sound-tag* and *tag-tag* network as described in the earlier section.

To evaluate the performance of the proposed system for audio lifelogging, 16 hours of audio was recorded continuously each day by a single subject using a lapel microphone attached to a recorder. This process is repeated for 3 days, resulting in a total of 48 hours of test data. Efforts were made to cover diverse sounds, ranging from indoor to outdoor activities. Each 16-hour audio recording is first fed to the feature extraction engine, followed by segmentation for event-boundary detection. Segmentation is performed using only three features - loudness, spectral centroid and spectral sparsity. This process results in a total of 2760 discrete events. To test annotation and retrieval performance, these segments were first manually annotated to obtain a ground truth. However, among these segments, most of the sounds are repetitive (for example, conversations taking place every few intervals or sounds of a car engine while going out multiple times in a day). In order to reduce the burden of annotation, the test dataset is trimmed to 62 distinct sounds, which were then tagged manually and compared to the automatically generated annotations.

The performance results for the tagging mechanism are documented in Table 6.1. The 3 highest-ranked tags obtained after automatic annotation are respectively compared to the tags assigned manually. One reason behind the relatively low per-

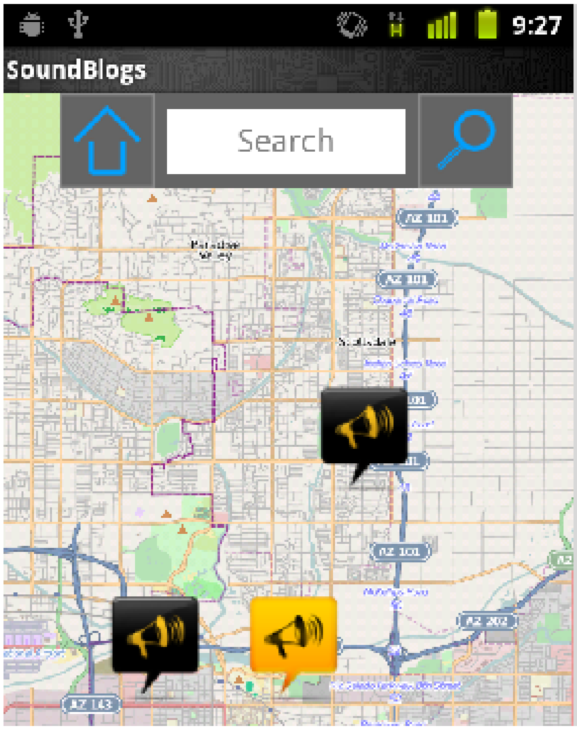


Figure 6.7: A Screenshot of the User Interface on an Android-Powered Smartphone.

formance is due to limitations of the current feature set in its inability to correctly identify male/female speakers as well as general speech conversations. To overcome this, the feature set used for annotation was augmented to include the first 15 MFCCs. The latter are routinely used in speech-related applications, justifying their choice in this study. The algorithm performs relatively better on this augmented feature set as evident from the results in Table 6.1. Since annotation and retrieval are two aspects of the same algorithm, the performance evaluation of the former can be extended to the latter, hence, no extra attempts are made to outline the retrieval performance separately.

6.2.1 Smartphone/Tablet-based User Interface

In order to create an intuitive and easy-to-use interface for lifelogging, a prototypical Android-based application, *SoundBlogs*, was developed. This application allows the user to either choose to continuously record audio for long durations (hours) or record short, interesting events (seconds). In the former case, at the end of the day, the user can plug his/her smartphone to a desktop computer application that automatically segments, annotates and archives this long recording to memory. While in the latter case, the application can be used for instant blogging and sharing interesting incidents or sounds with friends and colleagues. The user-interface is enhanced by capturing the current location of the user and displaying it as an icon on an interactive map. Selecting the icon prompts the user to listen to, provide a brief description, and finally archive or share the event.

Search or retrieval is enabled by prompting the user to enter a keyword related to the event. The application not only returns the event that matches the keyword, but also displays other events recorded on the same day along with their respective locations on a map. An example of search is shown in Figure 6.7. The yellow icon on the map indicates the event for which a match is obtained with the corresponding query keyword. Black icons represent other sound events not related to the keyword but recorded on the same day. Such an interface is reminiscent of, and aligns smoothly with, the concept of keeping a diary of personal information, which is one of the core aspects of any lifelogging application.

6.3 Summary

A complete framework covering feature extraction, segmentation, annotation and retrieval of long duration audio recordings in a lifelogging scenario was presented in this chapter. A conventional feature set was augmented with the MFCCs to account for the frequently occurring speech activities in the subject's daily life; and naturally led to a better performance. A prototype user-interface on an Android platform for smartphones or tablets showed how newer platforms and devices can be exploited to bring forth novel ways of lifelogging and visualization.

Chapter 7

LIFELOGGING: VIRTUAL PLATFORM MODELING TECHNIQUES

Virtual platforms (VP) allow software to be tested prior to silicon availability which reduces Time-To-Market (TTM) for a new product launch. Several products such as Synopsys Platform Architect [120], Bochs [121] and QEMU [122] are commonly used for this purpose. A virtual platform is typically created after the product architecture is fairly stable, which means limited time to use it prior to silicon availability. This design flow is better explained in Figure 7.1. The idea of starting virtual platform development right from the time when product concepts are being formed and architecture details are not fully available is explored in this chapter. This allows product ideas to be fully studied before key decisions are made, and enables this first virtual platform to gradually be used for product development. This can be seen in the apparent left-shift of the design flow in Figure 7.2, reducing the TTM even more.

SystemC/TLM2.0 or C/C++ based virtual platforms have been used extensively for modeling embedded devices [123, 124]. In [125], an MPEG-decoder was modeled using a modified version of QEMU [38], known as QEMU-SystemC. QEMU, along with QEMU-SystemC, provides a flexible and easy-to-use environment for instantiating virtual models of various devices and processors. In existing works, the primary focus has been on constructing virtual platforms for single devices such as an ASIC or an embedded mobile platform [126]. However, most of the present applications include interactions between multiple devices connected via the Internet, or commonly referred to as the Internet-of-Things (IOT). For example, a video tracking application involves a sensor for video capture, a server for analysis and communication between the two devices for complete operation. In addition, the devices are tightly coupled

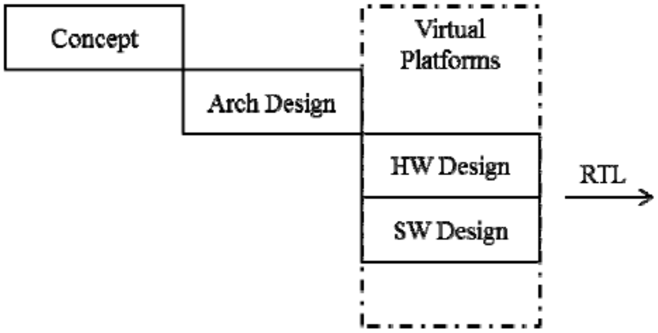


Figure 7.1: Current HW/SW Design Flow.

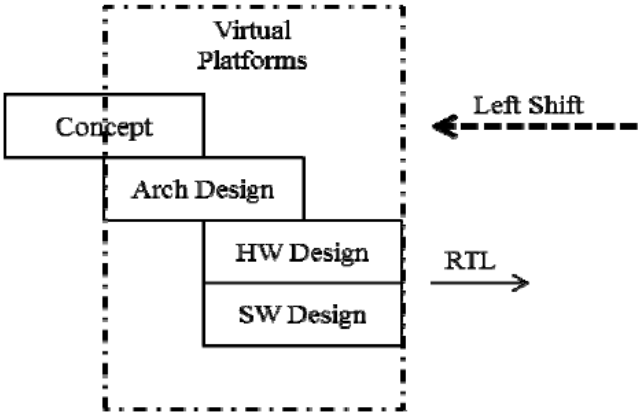


Figure 7.2: HW/SW Design Flow Using Virtual Platforms.

such that the design constraints and objectives of one device might affect the performance of the other. Hence, a joint optimization needs to be performed across all devices to ensure better overall performance.

7.1 Tools

The design of devices such as sensors or smartphones generally involves a main processor for generic tasks and a set of peripheral coprocessors to handle the com-

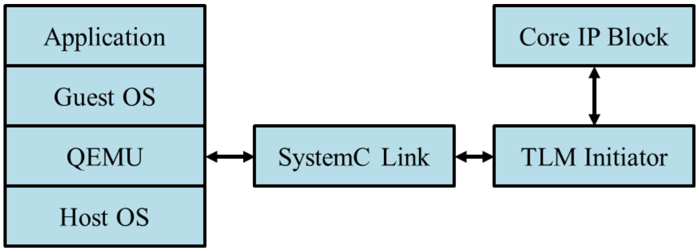


Figure 7.3: Architecture of a QEMU-SystemC Emulator.

putationally intensive tasks. The main processor runs an operating system such as GNU/Linux or Android, while device drivers are written to access the peripheral co-processors. QEMU [122] is an open-source virtualization and emulation tool. It can emulate entire systems based on x86, ARM, SPARC and other platforms along with their peripheral devices. It also allows for designers to write their own virtual devices, plug them in QEMU and evaluate their performance. Due to these features, QEMU is chosen here as the basic building block for creating virtual platforms.

SystemC and TLM2.0 [39] are two widely used industry standards for modeling hardware devices and communication interfaces in complex systems. Multiple levels of abstraction are offered, allowing designers to model abstractions ranging from a functional-level and blocking transport mechanism to a register-accurate, cycle-approximate and non-blocking mechanism. Although it is possible to model x86 and ARM instruction sets, the computational overhead of SystemC tremendously slows down the simulation (relative to QEMU). A variation of QEMU called QEMU-SystemC [125] is capable of running virtual hardware devices written in SystemC/TLM2.0 within QEMU. This combination enables high speed CPU modeling with SystemC accuracy for peripheral models.

The architecture for QEMU-SystemC is shown in Figure 7.3. At the base level, a SystemC link acts as a bridge attached to the PCI, AMBA or IO bus of the platform.

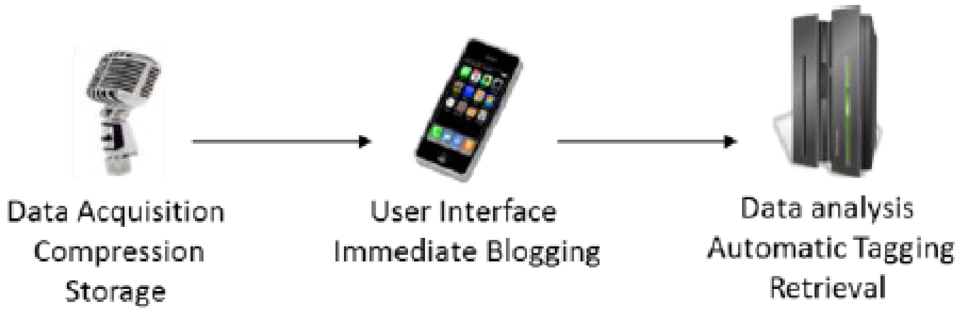


Figure 7.4: Components of the Virtual System.

The designers device is attached to this SystemC link. A read or write instruction to the virtual device is first transferred to the bridge, which further translates this instruction into a TLM2.0 compliant instruction and forwards it to the virtual device. This infrastructure is quite useful since it retains the fast execution speeds of QEMU as well as the design abstraction levels offered by SystemC/TLM2.0. QEMU-SystemC is well suited for sensor design, since sensors often involve the design of custom peripheral blocks or coprocessors. QEMU-SystemC facilitate the testing of such complex systems. Similarly, the Android emulator is another stable variation of QEMU capable of emulating a full-fledged Android smartphone. The three variants, QEMU, QEMU-SystemC and Android emulator are used in this study to create virtual platform models specific to a lifelogging application.

7.2 Virtual System Design

The virtual system includes a wearable recording device or sensor, a smartphone and a server, shown in Figure 7.4. Each component is assigned a set of specific tasks to be performed and design constraints it must satisfy. In this case, the sensor is responsible for continuous audio recording and compression. Likewise, the smartphone is responsible for providing the user with an intuitive user interface. The server is required to analyze the user-uploaded sound events and automatically annotate and archive them to a database. These individual components are modeled and instantiated using the aforementioned variants of QEMU on a single host machine. The host machine provides the interface for communication between these components. The implementation details for each of these components are addressed below.

The sensor is modeled as an x86 system running a GNU/Linux operating system on top of QEMU. Functionally, the sensor is responsible for compressing the incoming raw audio data and storing it. Here, the Ogg/Vorbis [127] standard is chosen as the compression algorithm due to its royalty and patent-free nature. Source code for such an encoder is freely available and is run on this virtual device without any modifications and additional coprocessors. Once compressed, the audio is stored in memory and the sensor awaits further instructions from the smartphone device for data transfer.

An Android emulator is used to model a smartphone capable of emulating GPS-based locations and wireless data transfer. Functionally, the smartphone must provide a smooth and intuitive user-interface for the audio blogging application. A user should be able to upload the recorded, compressed and stored audio clip in the sensor to his/her personal archive or publicly blog about it. It should allow the user to search through his/her archives for past recordings, thereby serving as a useful memory extension. Location plays an important role in categorizing these memories (clips), since sounds are often influenced by the surrounding environment.

A full-fledged GNU/Linux server on QEMU emulating an x86 platform is used for the server. This device is responsible for serving up web pages as well as performing a set of classification and retrieval algorithms on the user uploaded recordings. As is the case with the development of algorithms, they are first written in a high-level language such as Matlab for simulation and testing purposes and then ported to C/C++ for

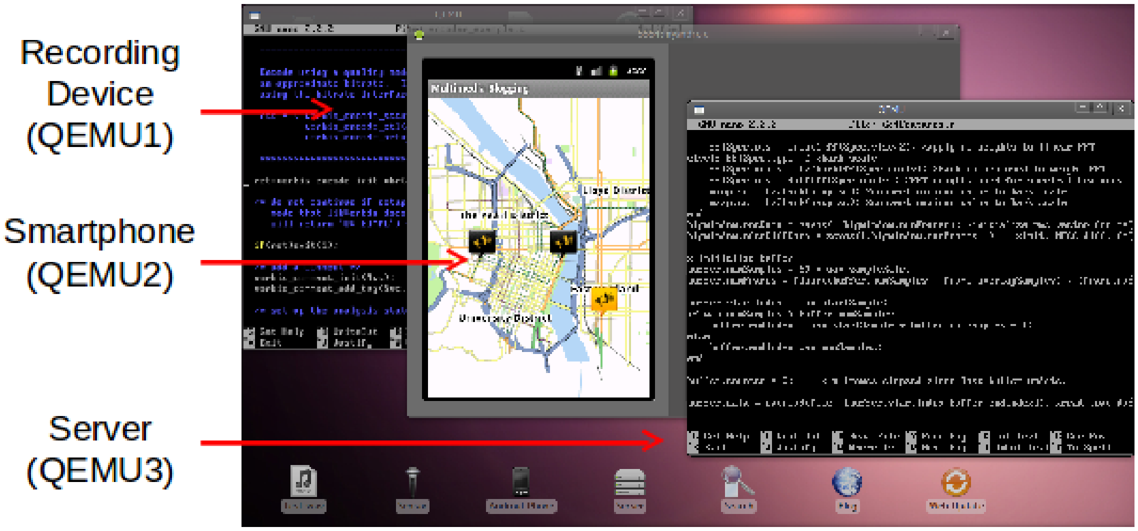


Figure 7.5: Screenshot of the Virtual System Framework on QEMU.

better speed and memory performance during production. To illustrate the ability of QEMU in handling this aspect, GNU/Octave is used on the server model to simulate these algorithms.

The virtual system comprising of the three devices instantiated separately on a single host machine is shown in Figure 7.5. The sequence of actions to perform blogging using this framework can be outlined as follows: (i) An audio clip is recorded in to the virtual sensor using an external or the host's in-built microphone. (ii) The Ogg/Vorbis encoder program is called to compress the clip and store it. (iii) This clip is then simultaneously uploaded to the virtual smartphone and the virtual server. (iv) The Android application on the smartphone displays this clip along with ad-hoc GPS coordinates as an icon on the map. (v) The server processes the data and automatically annotates it with relevant tags. (vi) Once the user decides to archive/publish this event, all the details, including the audio clip, GPS-based coordinates, date, time, tags, description, are packaged in to a single file and uploaded to the users blog website or personal archive.

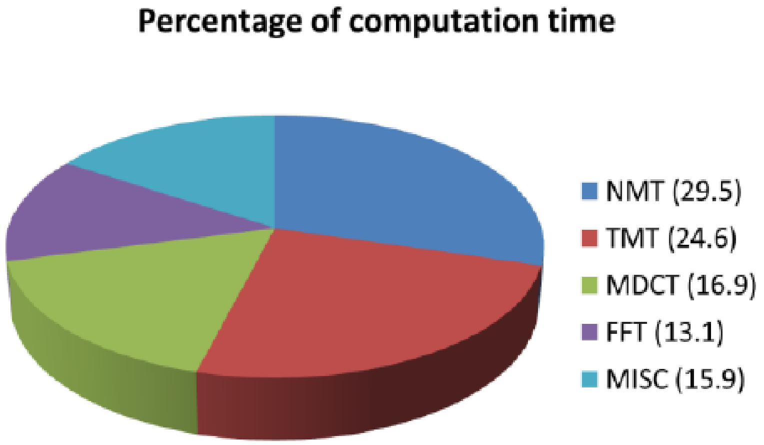


Figure 7.6: Time Profile of an Ogg/Vorbis Software Encoder.

7.3 Top-Down Design Methodology

7.3.1 Sensor

For the sensor, the objective is to build a non-obstructive, wearable device that can be attached to the lapel. This objective restricts the size of the device and consequently the size of the battery. The device is also required to record, compress and store audio for up to 24 hours without having to recharge the batteries. In order to satisfy this requirement, the device must operate at ultra-low power. This requires significant changes to be made to the compression algorithm in order to reduce the computational complexity as much as possible.

In order to perform such optimizations, first, the computationally intensive routines of an Ogg/Vorbis encoder must be identified. A time profile of the Ogg/Vorbis encoder and the sub-routines is shown in Figure 7.6. These routines include the Fast Fourier Transform (FFT), Modified Discrete Cosine Transform (MDCT), Tone-Masking Threshold (TMT), Noise-Masking Threshold (NMT) and miscellaneous op-

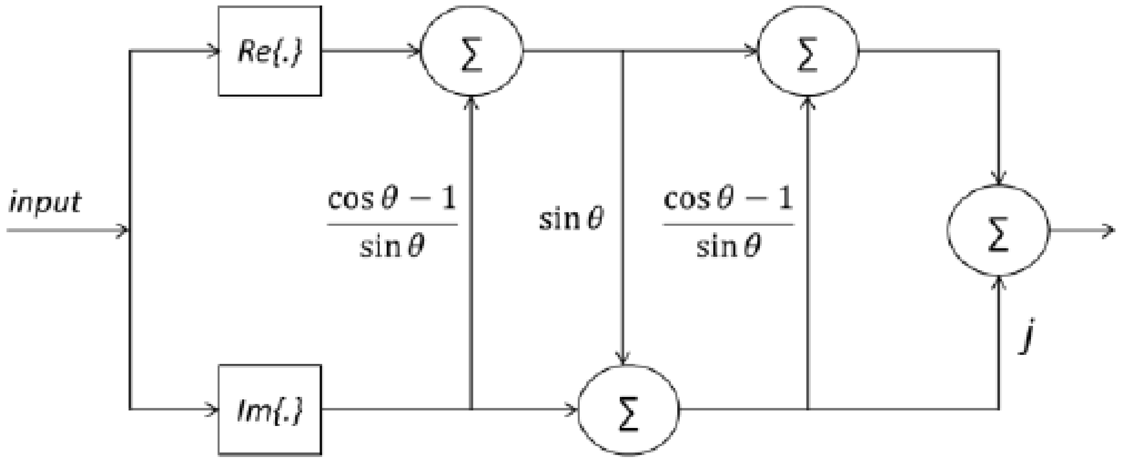

input
Re{.}
Im{.}
Σ
Σ
Σ
Σ
$\frac{\cos\theta - 1}{\sin\theta}$
$\sin\theta$
$\frac{\cos\theta - 1}{\sin\theta}$
j

Figure 7.7: Lattice Structure for Complex Multiplications.

erations related to Huffman encoding and packing the encoded data. Although TMT and NMT constitute 54.1% of the total time, they involve a large number of comparison and threshold operations which are inherently simple to implement and cannot be optimized further. The FFT and MDCT routines, taking 30% of the total time, rely on a significant amount of costly real or complex multiplication operations.

Based on this information, the next step is aimed at modifying the FFT and MDCT routines to reduce the overall complexity. The multiplication operations involved in computing the MDCT and FFT are based on the popular butterfly unit. In [128], a lifting-based method was derived to convert these butterfly units to lattice structures, as shown in Figure 7.7. This transformation allows for quantization and perfect reconstruction without considerable loss of precision. The trigonometric coefficients employed in these computations can be stored as dyadic rational numbers [129]. As a result, multiplications can now be implemented as a series of shift-add operations. The precision of raw audio samples and coefficients are set to 16-bit and 10-bit signed integers respectively. The reconstruction error based on these specifica-

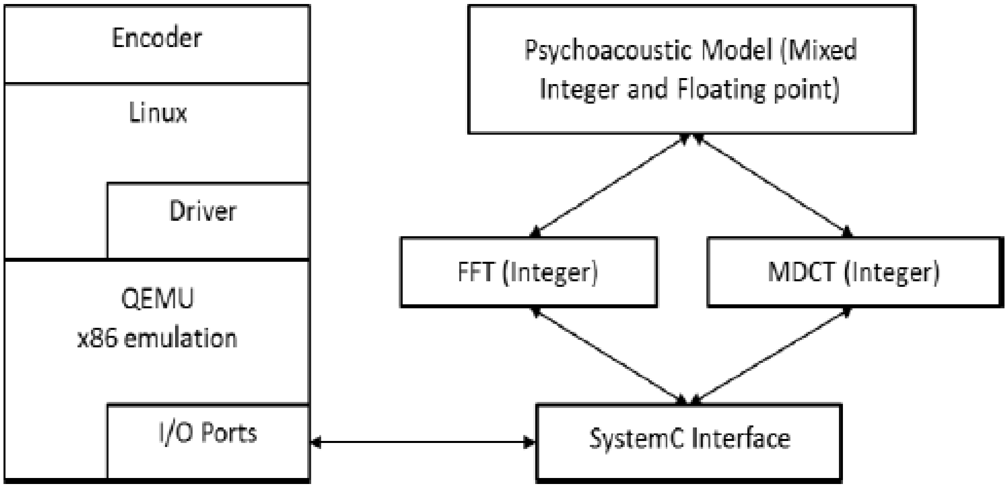


Figure 7.8: QEMU-SystemC Architecture for the Refined Sensor Model.

tions, as given in [128], is approximately -100 dB. This error is low enough to preserve the quality of recorded audio and not affect the quality of sound recognition.

Virtual hardware devices or coprocessors using SystemC/TLM2.0 were then developed to perform MDCT, FFT, TMT and NMT. The MDCT and FFT are implemented using integer-point arithmetic. The TMT and NMT combine to form the Psychoacoustic Model (PAM) and are implemented in mixed integer and floating-point arithmetic. At first, these devices are tested in a pure SystemC/TLM2.0 simulation environment. A traffic generator sends bursts of data to a generic router. The data is routed to the proper device, either one from FFT, MDCT or PAM, for further computations. A virtual device emulating a ROM is also designed to store sets of coefficients used in FFT and MDCT operations. These blocks are then plugged into QEMU-SystemC through the SystemC interface. The Ogg/Vorbis application code is modified to transfer these operations to the peripheral blocks. For this purpose, special device drivers are written on top of the GNU/Linux operating system. The QEMU-SystemC architecture is shown in Figure 7.8.

Table 7.1: Time Taken to Encode 25 Seconds of Raw Audio on Different Platforms.

Platform Type (x86-based)	Time (seconds)
Real PC	1.3
QEMU	2.1
QEMU-SystemC	20.4

To evaluate the efficiency of QEMU-SystemC in running complex applications, the Ogg/Vorbis encoder application is run on three different platforms a real PC, QEMU and QEMU-SystemC with coprocessors for FFT, MDCT and PAM. The time taken by each platform to encode 25 seconds of audio data at a rate of 64kbps is documented in Table 7.1. The time increases approximately by a factor of 15 on a QEMU-SystemC platform compared to a real PC. This can be attributed to the additional number of instructions required for the transfer of data between QEMU and the peripheral devices across the SystemC interface. However, the simulation time is still considerably low compared to simulating the entire system at cycle-accurate level on an FPGA. Thus, QEMU-SystemC facilitates rapid prototyping and validation at the cost of increasing the level of abstraction.

7.3.2 Smartphone

Location information available from the onboard GPS hardware is used in the Android application to enhance the interface. The application is ported from the QEMU-based Android emulator to an actual smartphone to enable real-time location capture. The application is also tested by various users and their feedback was used to further refine the interface and make it more user-friendly.

7.3.3 Server

The server model includes a fully functioning algorithm for automatic annotation and retrieval written in GNU/Octave. However, there is a large overhead in using such simulation tools in terms of speed and memory performance. Hence, the algorithms are ported to C++. The modified server model now closely resembles a real-world server. The algorithms are then deployed on an Amazon EC2 cloud-based server. This allows the actual smartphone application universal data access using its wireless capabilities. The system is updated with these refined models and the performance is evaluated again. The projected final system could be an implementation of the sensor on an FPGA device, a user application on a smartphone and the algorithms deployed on an actual server.

7.4 Concept Development Kit

Although the methods and results have been discussed specifically with respect to lifelogging using ambient sounds, the same framework can be extended towards the development of any application that involves multiple devices arranged in a similar configuration, i.e. a sensor or an array of sensors, a smartphone/PC and a server. Example applications include a medical device for continuously monitoring a users health or building a gesture recognition controller for mobile devices. The medical device or gesture recognition controllers stated here are essentially data acquisition devices and can be modeled as sensors. Coprocessors for acquiring and storing data can be modeled using SystemC/TLM2.0 and then plugged in to QEMU-SystemC. Similarly, a smartphone or a PC to present results to the user can be modeled using QEMU or the Android emulator. Finally, computationally intensive algorithms for analysis can be deployed on a server modeled using QEMU. Hence, using the same tools and framework presented here, one can model a variety of applications right at the concept stage.

Exploiting this similarity across different applications, a generic framework, Concept Development Kit (CDK), is proposed. The CDK includes SystemC and TLM2.0 for the design of custom hardware and QEMU-SystemC to evaluate their performance when integrated into a complex system. It also includes the Android emulator, along with templates and documentation, for designing user applications and interfaces. To model data processing servers with QEMU, the kit includes a number of different tools such as GNU/Octave and Python for deploying algorithms; Linux, Apache, MySQL and PHP (LAMP) for server administration and designing web applications such as blog websites or social media networks. The transfer and communication of data between these different QEMU instances is handled by an easy-to-use program based on the SSH protocol, also provided with the kit. Once the concept prototype is tested, the existing virtual platform can be further refined to become the main platform during production.

7.5 Summary

Using QEMU and QEMU-SystemC, virtual platform models were developed to simplify the design and implementation of a novel concept such as lifelogging. A top-down, iterative design methodology facilitated design in a rapid and parallel fashion. By taking the design objectives and constraints of different devices into consideration, the virtual platform model was jointly optimized for a better overall system performance. The possibility of refining and using the same virtual platforms at a later stage, during production, for example, effectively helps to reduce the TTM significantly. A CDK was proposed; this collection of tools provides an easy-to-use framework for design and development using this top-down methodology.

Chapter 8

SUMMARY

In this dissertation, context recognition using audio signals was studied for the purpose of improving human-machine interaction. Two scenarios were considered - (i) *active* interaction in emotion recognition and keyword detection, and (ii) *passive* interaction for lifelogging. Identifying contextual information, such as emotions or ambient sounds, is quite challenging owing to the numerous differences across speakers, environments and recording conditions. As a result, there is a strong requirement for efficient computational methods for feature extraction, representation and classification that can provide highly accurate estimates in real-time and across varying conditions.

A novel supervised method for feature extraction using LTMs was proposed for acoustic emotion recognition in Chapter 2. The proposed method, sRSM, learns discriminative, high-level features from the co-occurrence patterns among low-level descriptors. Experiments were conducted on multiple databases recorded across different languages, accents and cultures. Results show a significant improvement for categorical emotion recognition and valence discrimination. The latter is identified to be a hard task using only acoustic features, hence, the improvements obtained in this research are noteworthy. Based on further experiments, the proposed method was found to be highly suitable for long duration turns, which is a highly desirable property for current turn-based practices. Cross-corpus studies were conducted to evaluate the generalization ability of these features and the results were found to be quite promising. Software and FPGA implementations were provided to determine the feasibility for real-time applications.

The feature extraction framework was extended to multiple modalities in Chapter 3. Specifically, high-level features were extracted from facial expressions and spoken content. Individually, each source was identified to perform best at recognizing happy (face), sad (speech), and neutral (language). A multi-modal fusion was shown to retain these individual characteristics and improve the overall performance.

An articulation constrained learning method was proposed to perform emotion recognition using both acoustic and articulatory information in Chapter 4. A conventional L1-regularized logistic regression cost function was extended to jointly optimize two tasks - (i) emotion classification via logistic regression, and (ii) articulatory reconstruction via least squares regression. Experiments were performed to evaluate speaker dependent as well as independent emotion recognition performance on multiple databases.Significant improvements were obtained for valence classification of vowels */AA/*, */AE/*,*/IY/* and */UW/*. Incorporating articulatory constraints was shown to significantly improve the rate of recognizing happy emotions and decrease the misclassification rate between emotions with similar arousal characteristics, i.e. happy-angry or neutral-sad. The performance in a cross-corpus setting was observed to be almost similar to the within-corpus scenario.

A complete framework covering feature extraction, segmentation, annotation and retrieval of long duration audio recordings in a lifelogging scenario was presented Chapter 6. A conventional feature set was augmented with the MFCCs to account for the frequently occurring speech activities in the subject's daily life and improve the performance. Virtual platform models were developed to simplify the design and implementation of a novel concept such as lifelogging in Chapter 7. By taking the design objectives and constraints of different devices into consideration, the virtual platform model was jointly optimized for a better overall system performance.

The proposed LTM-based method for feature extraction ignored the temporal information in the process of extracting high-level topics. As shown by the limited success of HMM-based methods, there is still relevant information in the temporal domain, which if considered, could be helpful towards improving the performance. A possible direction for future work could involve combining the RSM and HMM in a single learning framework in order to model the temporal dependencies between acoustic words. Additionally, the unsupervised dictionary learning process could be extended to account for label information and obtain more discriminative bag-of-words representations. Related to the effects of articulation on emotion recognition, the experiments conducted in this dissertation were limited to peripheral vowels, however, the ACL method itself is not restricted to vowels. Future work could be directed towards evaluating the performance on other vowels and consonants, and, build a generic learning framework that can readily combine with existing real-world applications.

REFERENCES

[1] A. M. Turing, "Computing machinery and intelligence," *Mind*, pp. 433–460, 1950.

[2] E. Ackerman, "Google gets in your face [2013 tech to watch]," *IEEE Spectrum*, vol. 50, no. 1, pp. 26–29, 2013.

[3] B. N. Schilit, N. Adams, R. Gold, M. M. Tso, and R. Want, "The PARCTAB mobile computing system," in *Proceedings of the 4th Workshop on Workstation Operating Systems.* IEEE, 1993, pp. 34–39.

[4] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," *Technical Report TR2000-381*, 2000.

[5] A. Chen, R. R. Muntz, S. Yuen, I. Locher, S. Sung, and M. B. Srivastava, "A support infrastructure for the smart kindergarten," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 49–57, 2002.

[6] J. J. Magee, M. Betke, J. Gips, M. R. Scott, and B. N. Waber, "A human–computer interface using symmetry between eyes to detect gaze direction," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 38, no. 6, pp. 1248–1261, 2008.

[7] H. Yan and T. Selker, "Context-aware office assistant," in *Proceedings of the 5th International Conference on Intelligent User Interfaces.* New Orleans: ACM, January 2000, pp. 276–279.

[8] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[9] C.-C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayanan, "Emotion recognition using a hierarchical binary decision tree approach," *Speech Communication*, vol. 53, no. 9, pp. 1162–1171, 2011.

[10] E. Mower, M. J. Mataric, and S. Narayanan, "A framework for automatic human emotion classification using emotion profiles," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1057–1070, 2011.

[11] A. Metallinou, S. Lee, and S. Narayanan, "Decision level combination of multiple modalities for recognition and analysis of emotional expression," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2010, pp. 2462–2465.

[12] F. Eyben, M. Wollmer, and B. Schuller, "OpenEARintroducing the Munich open-source emotion and affect recognition toolkit," in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction and Workshops.* IEEE, 2009, pp. 1–6.

[13] T. L. Nwe, S. W. Foo, and L. C. De Silva, "Speech emotion recognition using hidden markov models," *Speech communication*, vol. 41, no. 4, pp. 603–623, 2003.

[14] M. M. El Ayadi, M. S. Kamel, and F. Karray, "Speech emotion recognition using gaussian mixture vector autoregressive models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, pp. 957–960.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[16] R. Salakhutdinov and G. E. Hinton, "Replicated Softmax: an Undirected Topic Model." in *Neural Information Processing Systems*, vol. 22, Lake Tahoe, 2009, pp. 1607–1614.

[17] M. Shah, L. Miao, C. Chakrabarti, and A. Spanias, "A speech emotion recognition framework based on latent Dirichlet allocation: Algorithm and FPGA implementation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.* Vancouver: IEEE, June 2013, pp. 2553–2557.

[18] M. Shah, C. Chakrabarti, and A. Spanias, "A multi-modal approach to emotion recognition using undirected topic models," in *Proceedings of the IEEE International Symposium on Circuits and Systems.* Melbourne: IEEE, June 2014, pp. 754–757.

[19] M. Shah, C. Chakrabarti, and A. Spanias, "Within and cross-corpus speech emotion recognition using latent topic model-based features," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, pp. 1–17, January 2015.

[20] D. Erickson, O. Fujimura, and B. Pardo, "Articulatory correlates of prosodic control: Emotion and emphasis," *Language and Speech*, vol. 41, no. 3-4, pp. 399–417, 1998.

[21] M. Nordstrand, G. Svanfeldt, B. Granström, and D. House, "Measurements of articulatory variation in expressive speech for a set of swedish vowels," *Speech Communication*, vol. 44, no. 1, pp. 187–196, 2004.

[22] S. Lee, S. Yildirim, A. Kazemzadeh, and S. Narayanan, "An articulatory study of emotional speech production." in *Proceedings of INTERSPEECH*, 2005, pp. 497–500.

[23] D. R. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[24] S. Parlak and M. Saraclar, "Spoken term detection for turkish broadcast news," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2008, pp. 5244–5247.

[25] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proceedings of the 30th Annual International ACM SIGIR conference on Research and development in Information Retrieval.* ACM, 2007, pp. 615–622.

[26] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.* IEEE, 1989, pp. 627–630.

[27] R. C. Rose and D. B. Paul, "A hidden markov model based keyword recognition system," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.* IEEE, 1990, pp. 129–132.

[28] J. Wilpon, L. Miller, and P. Modi, "Improvements and applications for key word recognition using hidden markov modeling techniques," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.* IEEE, 1991, pp. 309–312.

[29] M.-C. Silaghi and H. Bourlard, "Iterative posterior-based keyword spotting without filler models," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop.* IEEE, 1999, pp. 213–216.

[30] M.-C. Silaghi, "Spotting subsequences matching an hmm using the average observation probability criteria with application to keyword spotting," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 20. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 1118.

[31] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuous speech recognition with context-dependent dbn-hmms," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2011, pp. 4688–4691.

[32] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2014, pp. 4087–4091.

[33] V. Bush, "As we may think," *The Atlantic Monthly*, vol. 176, no. 1, pp. 101–108, 1945.

[34] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE Multimedia*, vol. 3, no. 3, pp. 27–36, 1996.

[35] D. P. Ellis and K. Lee, "Accessing minimal-impact personal audio archives," *IEEE Multimedia*, vol. 13, no. 4, pp. 30–38, 2006.

[36] G. Wichern, H. Thornburg, B. Mechtley, A. Fink, K. Tu, and A. Spanias, "Robust multi-features segmentation and indexing for natural sound environments," in *International Workshop on Content-Based Multimedia Indexing.* Bordeaux: IEEE, June 2007, pp. 69–76.

[37] G. Wichern, J. Xue, H. Thornburg, B. Mechtley, and A. Spanias, "Segmentation, indexing, and retrieval for environmental and natural sounds," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 688–707, 2010.

[38] "QEMU," 2015. [Online]. Available: http://www.qemu.org

[39] "SystemC," 2015. [Online]. Available: http://www.systemc.org

[40] M. Shah, B. Mears, C. Chakrabarti, and A. Spanias, "Lifelogging: Archival and retrieval of continuously recorded audio using wearable devices," in *Proceedings of IEEE Conference on Emerging Signal Processing Applications.* Las Vegas: IEEE, January 2012, pp. 99–102.

[41] M. Shah, and B. Mears, and C. Chakrabarti, and A. Spanias, "A top-down design methodology using virtual platforms for concept development," in *Proceedings of the International Symposium on Quality Electronic Design.* Santa Clara: IEEE, March 2012, pp. 444–450.

[42] C. M. Lee and S. S. Narayanan, "Toward detecting emotions in spoken dialogs," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 2, pp. 293–303, 2005.

[43] L. Vidrascu and L. Devillers, "Detection of real-life emotions in call centers." in *Proceedings of INTERSPEECH*, 2005, pp. 1841–1844.

[44] R. Cowie and R. R. Cornelius, "Describing the emotional states that are expressed in speech," *Speech communication*, vol. 40, no. 1, pp. 5–32, 2003.

[45] S. Steidl, "Automatic classification of emotion-related user states in spontaneous children's speech," *Ph.D Thesis*, 2009.

[46] S. Narayanan and P. G. Georgiou, "Behavioral signal processing: Deriving human behavioral informatics from speech and language," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1203–1233, 2013.

[47] P. Ekman, D. Matsumoto, and W. V. Friesen, "Facial expression in affective disorders," *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*, vol. 2, pp. 331–342, 1997.

[48] G. I. Roisman, J. L. Tsai, and K.-H. S. Chiang, "The emotional integration of childhood experience: physiological, facial expressive, and self-reported emotional response during the adult attachment interview." *Developmental psychology*, vol. 40, no. 5, p. 776, 2004.

[49] J. F. Cohn and E. Z. Tronick, “Mother–infant face-to-face interaction: The sequence of dyadic states at 3, 6, and 9 months.” *Developmental Psychology*, vol. 23, no. 1, p. 68, 1987.

[50] M. Pantic, A. Pentland, A. Nijholt, and T. S. Huang, “Human computing and machine understanding of human behavior: a survey,” *Artifical Intelligence for Human Computing*, pp. 47–71, 2007.

[51] B. Schuller, A. Batliner, S. Steidl, and D. Seppi, “Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge,” *Speech Communication*, vol. 53, no. 9, pp. 1062–1087, 2011.

[52] D. Wu, T. D. Parsons, E. Mower, and S. Narayanan, “Speech emotion estimation in 3d space,” in *Proceedings of the International Conference on Multimedia and Expo*. IEEE, 2010, pp. 737–742.

[53] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, “A survey of affect recognition methods: Audio, visual, and spontaneous expressions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, 2009.

[54] B. Schuller and F. Weninger, “Ten recent trends in computational paralinguistics,” *Cognitive Behavioural Systems*, pp. 35–49, 2012.

[55] C. Busso, S. Lee, and S. Narayanan, “Analysis of emotionally salient aspects of fundamental frequency for emotion detection,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 582–596, 2009.

[56] K. R. Scherer, “Vocal affect expression: a review and a model for future research.” *Psychological bulletin*, vol. 99, no. 2, p. 143, 1986.

[57] C. Gobl and A. Nı Chasaide, “The role of voice quality in communicating emotion, mood and attitude,” *Speech communication*, vol. 40, no. 1, pp. 189–212, 2003.

[58] H. Teager, “Some observations on oral air flow during phonation,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 5, pp. 599–601, 1980.

[59] G. Zhou, J. H. Hansen, and J. F. Kaiser, “Classification of speech under stress based on features derived from the nonlinear Teager energy operator,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. IEEE, 1998, pp. 549–552.

[60] M. El Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[61] B. Schuller, S. Steidl, and A. Batliner, “The INTERSPEECH 2009 emotion challenge.” in *Proceedings of INTERSPEECH*, 2009, pp. 312–315.

[62] B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic, "Avec 2011–the first international audio/visual emotion challenge," *Affective Computing and Intelligent Interaction*, pp. 415–424, 2011.

[63] B. Schuller, M. Valster, F. Eyben, R. Cowie, and M. Pantic, "Avec 2012: the continuous audio/visual emotion challenge," in *Proceedings of the 14th ACM International Conference on Multimodal Interaction*. ACM, 2012, pp. 449–456.

[64] B. Schuller, B. Vlasenko, F. Eyben, G. Rigoll, and A. Wendemuth, "Acoustic emotion recognition: A benchmark comparison of performances," in *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE, 2009, pp. 552–557.

[65] M. Wöllmer, N. Klebert, and B. Schuller, "Switching linear dynamic models for recognition of emotionally colored and noisy speech," *ITG-Fachbericht-Sprachkommunikation 2010*, 2010.

[66] M. Wollmer, B. Schuller, F. Eyben, and G. Rigoll, "Combining long short-term memory and dynamic bayesian networks for incremental emotion-sensitive artificial listening," *IEEE Jornal of Selected Topics in Signal Processing*, vol. 4, no. 5, pp. 867–881, 2010.

[67] M. Wöllmer, F. Eyben, S. Reiter, B. Schuller, C. Cox, E. Douglas-Cowie, and R. Cowie, "Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies." in *Proceedings of INTERSPEECH*, 2008, pp. 597–600.

[68] W. Duch, J. Biesiada, T. Winiarski, K. Grudziński, and K. Grabczewski, "Feature selection based on information theory filters," *Neural Networks and Soft Computing*, pp. 173–178, 2003.

[69] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian, "Feature selection using principal feature analysis," in *Proceedings of the 15th International Conference on Multimedia*. ACM, 2007, pp. 301–304.

[70] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 3687–3691.

[71] O.-W. Kwon, K. Chan, J. Hao, and T.-W. Lee, "Emotion recognition by speech signals." in *Proceedings of INTERSPEECH*, 2003.

[72] J. Nicholson, K. Takahashi, and R. Nakatsu, "Emotion recognition in speech using neural networks," *Neural computing & applications*, vol. 9, no. 4, pp. 290–296, 2000.

[73] B. Schuller, A. Batliner, D. Seppi, S. Steidl, T. Vogt, J. Wagner, L. Devillers, L. Vidrascu, N. Amir, L. Kessous *et al.*, "The relevance of feature type for the automatic classification of emotional user states: low level descriptors and functionals." in *Proceedings of INTERSPEECH*, vol. 2007, 2007, pp. 1–4.

[74] S. Pan, J. Tao, and Y. Li, "The CASIA audio emotion recognition method for audio/visual emotion challenge 2011," *Affective Computing and Intelligent Interaction*, pp. 388–395, 2011.

[75] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proceedings of the International Conference on Ubiquitous Computing.* ACM, 2008, pp. 10–19.

[76] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines." in *Neural Information and Processing Systems*, 2012, pp. 2231–2239.

[77] C. Wang, D. Blei, and F.-F. Li, "Simultaneous image classification and annotation," in *Proceedings of Computer Vision and Pattern Recognition.* IEEE, 2009, pp. 1903–1910.

[78] D. Liu and T. Chen, "Unsupervised image categorization and object localization using topic models and correspondences between images," in *Proceedings of the International Conference on Computer Vision.* IEEE, 2007, pp. 1–7.

[79] S. Kim, S. Narayanan, and S. Sundaram, "Acoustic topic model for audio information retrieval," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.* IEEE, 2009, pp. 37–40.

[80] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, 1990.

[81] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGIR Conference on Research and development in Information Retrieval.* ACM, 1999, pp. 50–57.

[82] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[83] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[84] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

[85] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK book.* Entropic Cambridge Research Laboratory Cambridge, 1997, vol. 2.

[86] M. J. Beal, "Variational algorithms for approximate Bayesian inference," 2003.

[87] A. Stuhlsatz, J. Lippel, and T. Zielke, "Feature extraction with deep neural networks by a generalized discriminant analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 4, pp. 596–608, 2012.

[88] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, and B. Schuller, "Deep neural networks for acoustic emotion recognition: Raising the benchmarks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2011, pp. 5688–5691.

[89] S. Press and S. Wilson, "Choosing between logistic regression and discriminant analysis," *Journal of the American Statistical Association*, vol. 73, no. 364, pp. 699–705, 1978.

[90] M. Pohar, M. Blas, and S. Turk, "Comparison of logistic regression and linear discriminant analysis: a simulation study," *Metodolski Zvezki*, vol. 1, no. 1, pp. 143–161, 2004.

[91] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," *Proceedings of GSCL*, pp. 31–40, 2009.

[92] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of German emotional speech." in *Proceedings of INTERSPEECH*, vol. 5, 2005, pp. 1517–1520.

[93] I. S. Engberg, A. V. Hansen, O. Andersen, and P. Dalsgaard, "Design, recording and verification of a danish emotional speech database." in *Proceedings of Eurospeech*, 1997.

[94] M. Grimm, K. Kroschel, and S. Narayanan, "The Vera am Mittag German audio-visual emotional speech database," in *Proceedings of the International Conference on Multimedia and Expo.* IEEE, 2008, pp. 865–868.

[95] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schroder, "The Semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent," *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 5–17, 2012.

[96] A. Batliner, S. Steidl, and E. Nöth, "Releasing a thoroughly annotated and processed spontaneous emotional database: the FAU Aibo emotion corpus," in *Proceedings of LREC*, 2008, pp. 28–31.

[97] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[98] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[99] M. Glodek, S. Tschechne, G. Layher, M. Schels, T. Brosch, S. Scherer, M. Kächele, M. Schmidt, H. Neumann, G. Palm *et al.*, "Multiple classifier systems for the classification of audio-visual emotional states," *Affective Computing and Intelligent Interaction*, pp. 359–368, 2011.

[100] B. Schuller, B. Vlasenko, F. Eyben, M. Wollmer, A. Stuhlsatz, A. Wendemuth, and G. Rigoll, "Cross-corpus acoustic emotion recognition: variances and strategies," *IEEE Transactions on Affective Computing*, vol. 1, no. 2, pp. 119–131, 2010.

[101] D. Neiberg, P. Laukka, and H. A. Elfenbein, "Intra-, inter-, and cross-cultural classification of vocal affect." in *Proceedings of INTERSPEECH*, 2011, pp. 1581–1584.

[102] L. Devillers, C. Vaudable, and C. Chastagnol, "Real-life emotion-related states detection in call centers: a cross-corpora study." in *Proceedings of INTERSPEECH*, 2010, pp. 2350–2353.

[103] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *IEEE Computational Science & Engineering*, vol. 5, no. 1, pp. 46–55, 1998.

[104] G. Caridakis, G. Castellano, L. Kessous, A. Raouzaiou, L. Malatesta, S. Asteriadis, and K. Karpouzis, "Multimodal emotion recognition from expressive faces, body gestures and speech," *Artificial intelligence and innovations: From theory to applications*, pp. 375–388, 2007.

[105] N. Sebe, I. Cohen, and T. S. Huang, "Multimodal emotion recognition," *Handbook of Pattern Recognition and Computer Vision*, vol. 4, pp. 387–419, 2005.

[106] A. Katsamanis, M. Black, P. G. Georgiou, L. Goldstein, and S. Narayanan, "Sailalign: Robust long speech-text alignment," in *Proceedings of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, 2011.

[107] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," 2001.

[108] A. Y. Ng, "Feature selection, l1 vs. l2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 78.

[109] S.-I. Lee, H. Lee, P. Abbeel, and A. Y. Ng, "Efficient l1 regularized logistic regression," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 401.

[110] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2006, pp. 801–808.

[111] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "A comparison of optimization methods and software for large-scale l1-regularized linear classification," *The Journal of Machine Learning Research*, vol. 11, pp. 3183–3234, 2010.

[112] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2008.

[113] F. Eyben, A. Batliner, B. Schuller, D. Seppi, and S. Steidl, "Cross-corpus classification of realistic emotions–some pilot experiments," in *Proceedings of the International Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect, Valetta*, 2010, pp. 77–82.

[114] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett, "The darpa 1000-word resource management database for continuous speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1988, pp. 651–654.

[115] D. Povey, A. Ghoshal, and et al., "The Kaldi speech recognition toolkit," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2011.

[116] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean *et al.*, "On rectified linear units for speech processing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 3517–3521.

[117] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition." in *Proceedings of INTERSPEECH*, 2013, pp. 2365–2369.

[118] J. Keshet, D. Grangier, and S. Bengio, "Discriminative keyword spotting," *Speech Communication*, vol. 51, no. 4, pp. 317–329, 2009.

[119] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong, "MyLifeBits: fulfilling the Memex vision," in *Proceedings of the 10th International Conference on Multimedia*. ACM, 2002, pp. 235–238.

[120] "Synopsys," 2015. [Online]. Available: http://www.synopsys.com

[121] "Bochs," 2015. [Online]. Available: http://bochs.sourceforge.net

[122] F. Bellard, "QEMU, a fast and portable dynamic translator." in *USENIX Annual Technical Conference, FREENIX Track*, 2005, pp. 41–46.

[123] A. Dion, E. Boutillon, V. Calmettes, and E. Liegon, "A flexible implementation of a Global Navigation Satellite System (GNSS) receiver for on-board satellite navigation," in *Design and Architectures for Signal and Image Processing (DASIP)*. IEEE, 2010, pp. 48–53.

[124] K. Grüttner, F. Oppenheimer, W. Nebel, F. Colas-Bigey, and A.-M. Fouilliart, "Systemc-based modelling, seamless refinement, and synthesis of a jpeg 2000 decoder," in *Proceedings of the Conference on Design, Automation and Test in Europe.* ACM, 2008, pp. 128–133.

[125] M. Monton, A. Portero, M. Moreno, B. Martinez, and J. Carrabina, "Mixed SW/SystemC SOC emulation framework," in *IEEE International Symposium on Industrial Electronics.* Vigo: IEEE, June 2007, pp. 2338–2341.

[126] S. Abdi, Y. Hwang, L. Yu, H. Cho, I. Viskic, and D. Gajski, "Embedded system environment: A framework for tlm-based design and prototyping," in *21st IEEE International Symposium on Rapid System Prototyping*, June 2010, pp. 1–7.

[127] "Ogg/Vorbis," 2015. [Online]. Available: http://www.xiph.org/vorbis/

[128] S. Oraintara, Y.-J. Chen, and T. Q. Nguyen, "Integer fast Fourier transform," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 607–618, 2002.

[129] T. D. Tran, "The BinDCT: Fast multiplierless approximation of the DCT," *IEEE Signal Processing Letters*, vol. 7, no. 6, pp. 141–144, 2000.