UNIVERSIDAD
DE ANTIOQUIA

Robust Automatic Speech Recognition

Luis Felipe Parra Gallego

Tesis de maestría presentada para optar al título de Magíster en Ingeniería de Telecomunicaciones

Director
Prof. Dr.-Ing Juan Rafael Orozco Arroyave

Asesor
MSc. Tomas Arias Vergara

Universidad de Antioquia
Facultad de Ingeniería
Maestría en Ingeniería de Telecomunicaciones
Medellín, Antioquia, Colombia
2022

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Robust Automatic Speech Recognition

UNIVERSIDAD
DE ANTIOQUIA
1803

Research work for the Master's degree in Telecommunications Engineering.

Luis Felipe Parra Gallego

Director: Prof. Dr.-Ing. Juan Rafael Orozco Arroyave
Advisor: M.Sc. Tomás Arias Vergara

Faculty of Engineering

Department of Electronic Engineering and Telecommunications

University of Antioquia

Acknowledgments

This work is mainly dedicated to my parents Edgar Parra López and María Lucía Gallego who, with their patience, support and unconditional love have allowed me to accomplish this great achievement. Thanks to them for installing in me the necessary values and foundations so that, with effort, I can continue on the road to success. I also want to thank my family for giving me their advice and words of encouragement throughout this process to be a better person and somehow accompany me in all my dreams and goals.

I would like to thank my colleagues and friends Cristian Ríos, Felipe Orlando and Daniel Escobar because they always supported me in my personal and professional development process, especially to Daniel who was always there in difficult moments. They also helped me in the development of this work. I would like to extend my thanks to my others colleages Nicanor García, Camilo Vásquez, Guberney Muñetón, Paula Pérez, Luis Gómez and Surley Berrío who, in some way, contributed to my education.

Likewise, my thanks to the company Konecta Group for financing both the master's program and my personal expenses. I also want to express my gratitude to my bosses Diana Velasquez and Andrés Maya for opening the doors of the company and allowing me to continue this process.

I would like to offer my greatest and most sincere thanks to my director Juan Rafael Orozco and my co-director Tomás Arias Vergara who with effort and determination were able to guide me in the development of this project. With their wisdom and knowledge, they have had a positive impact on both personal and professional aspects and for that I want to give them a special thanks.

Finally, thanks to the University of Antioquia and Konecta Group who financed this master's thesis through the grants #PI2019-24110 and #PRG2020-34068.

Abstract

In contact center organizations, customer satisfaction (CS) analysis is an important issue since the organization's reputation is strongly impacted by the customer's perception of the quality of service (QoS) provided. Service agents must provide exceptional service for the continual growth of the organization in today's dynamic market. In order to improve the service, these companies have human experts to evaluate the QoS. This practice is commonly based on the customer's opinion of the service after the conversation with the agent. However, such practice has two main disadvantages: 1) double cost and effort, i.e., human experts are needed to answer the calls as well as to evaluate them; and 2) only a small sample of the total number of calls is rated due to human limitations. Given these difficulties, these organizations have promoted research into the development of different systems based on acoustic and linguistic analyses that help to automatically evaluate CS. The acoustic-based system detects abnormal changes on the speech signal such as: poorly-articulated speech, increase in speech rate, increase in voice volume, and others. The linguistic-based system searches for keywords that reflect satisfaction/dissatisfaction. This approach requires an Automatic Speech Recognition (ASR) system to convert the speech signal into a text transcriptions. The ASR system must be designed in such a way that its performance is minimally dependent of the acoustic conditions. This thesis proposes a methodology to robustly recognize speech in non-controlled acoustic conditions using recordings collected by a call center. It also proposes a methodology to recognize emotion from speech & to evaluate CS based on acoustic and linguistic analysis. The acoustic features include articulation, prosody and phonation features. The linguistic features consist of word embeddings extracted from the transcriptions generated by the proposed ASR system. Deep learning approaches are considered for both speech recognition and CS evaluation and they are compared with traditional techniques.

Contents

Chapter 1

Introduction

This section first explains the motivation of this research work. Then, it describes the state-of-the-art and it explains the general and specific objectives of this study. Finally, it states the research problem and it presents the contribution of this thesis.

1.1 Motivation

Each day millions of calls that are answered in call centers are recorded and stored in storage centrals due to several regulations. The recordings are used for different purposes like processing and analysis, in order to improve the quality of the service. A common practice consists of listening to and evaluating interactions between business advisors and customers. This procedure is usually done by a human being who evaluates the service by randomly taking samples from the total calls. During the quality of service (QoS) evaluation, it is rated whether the business advisor resolved the customer's problem or need, whether s(he) performed it efficiently and timely, whether s(he) did not raise her/his voice tone and volume, and additionally s(he) answered calmly, whether customer got angry, etc [1]. However, this procedure has 2 main disadvantages: (1) There is an increased cost for rating the calls; answering the calls and evaluating them. (2) Only a few samples over the total calls are evaluated, so it is not possible to pick up all critical calls that could help in improving the QoS [1].

Several automatic systems designed to rate QoS in call centers have been released. Generally these systems are based on acoustic and/or linguistic analyses. On the one hand, the acoustic-based system detects abnormal changes on the speech signal such as: poorly-articulated speech, increase in speech rate, increase in voice volume, and others. It has been shown that these systems are suitable to assist call center managers in monitoring and optimizing the service provided by the agent. They can potentially detect the emotional state of agents and/or customers and hence provide a QoS index. However, this analysis is quite complex because it must deal with acoustic conditions of recording, environmental noise, acoustic differences between speakers, among others. On the other hand, the linguistic-based system is based on the textual content of the conversation between agents and customers as well as customers' opinions after the conversations (i.e., analysis in voicemails). Nonetheless, this other approach requires the text transcription of the spoken conversation in order to assign an accurate and efficient rate to the conversation. Some of the features that are computed upon the text and are related to QoS include keywords, key sentences, number and types of hesitations, and others. Thus a system of this nature consists of a speech to text converter to transcribe the calls and a text analyzer to compute the feature previously mentioned. From experience, textual analysis performs better CS estimation than acoustic analysis since there are words in voicemails that directly reflect the satisfaction/dissatisfaction of the service provided. That is, even with some perturbation in the text transcription (such as stop word deletions), the linguistic-based system can still evaluate CS by searching for keywords. In contrast, the acoustic-based systems are sensitive to changes in the acoustic conditions of the recordings. Regardless of the approach used (i.e. acoustic, linguistic or a combination of them), they allow to automatically analyze the 100% answered calls. Additionally, all interactions can be discriminated between good and bad service.

ASR systems are the natural alternative to address the problem of automatic QoS evaluation based on text analysis. The goal of an ASR is to efficiently and accurately convert a speech signal into its corresponding text transcription. An ASR system should be designed such that its performance does not depend on different conditions like the microphone, the accent of the speaker, acoustic conditions, and others [2]. Also, an appropriate technique is essential for achieving high performance; typical ASR is basically developed following 2 approaches: (1) Techniques based on traditional machine learning (ML) using Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM), or (2) Techniques based on Deep Neural Networks (DNN). The first approach is a standard tool for ASR systems. These systems are simpler and their computational complexity is low [3], [4]. On the other hand, the second approach plays an important role in current ASR systems. Recent advances have shown this technique to be promising [4]. Additionally, an ASR based on DNN has many advantages compared to traditional approaches. For instance, DNN can model the acoustic features of speech signal using less parameters than some ML-based approaches. Additionally, a DNN only needs one model for representing all sounds to be recognized, while ML models need as many individual models as sounds to be recognized [5].

The performance of a textual-based system to automatically rate QoS depends on the accuracy in transcription. This means transcriptions with errors could produce bad interpretations for the QoS evaluation system. For instance, if a customer claims disagreement by saying “I did not have a good service” and the recognizer omits the word “not”, it would completely change the rating result. Although ASR systems have high performance under ideal conditions, there are many factors that reduce the ASR performance, such as speaker’s acoustic trait (health condition and emotional state), communication channel (microphone and sound card), environmental noise, and others. Many works in ASR systems have been carried out for reducing harmful effects produced by noise. Improving robustness over this phenomenon in ASR systems would lead to raise their performance. Comparisons among different engines based on different techniques have been shown that those based on DNN are more robust against non-controlled acoustic environments. Using this technique, relative improvements of up to 7.5% word error rate (WER) have been reported [4].

Thus, in the last decade, the most important technique in speech recognition has been HMM-DNN hybrid models. The DNN-based acoustic model has shown significant improvements through the investigation of several network topologies [6]. Besides, end-to-end ASR systems are becoming popular due to their capability to be totally optimized (acoustic and language model at the same time). However, these systems suffer from the problem in which redundant generators repeat and importance symbols vanish [6]. Although an HMM-DNN-based ASR can not be totally optimized, it has the advantage of stable processing to estimate phoneme states on a frame basis and each component can be intervened separately [6]. For this reason, hybrid models are used to carry out this research work.

This master thesis also proposes a methodology to robustly recognize emotion and to estimate the CS based on acoustic and linguistic analyses. The first step consists of a exhaustive search of acoustic features to find the most effective ones to detect emotional states from speech. Then, the best acoustic features are combined with textual features to improve the CS estimation.

1.2 State-of-the-art

This section describes related work on speech recognition and emotion recognition from speech & automatic customer satisfaction estimation.

1.2.1 Automatic speech recognition

Several techniques have been proposed to model acoustic features in an ASR system. The two most common approaches used nowadays are those based on Hidden Markov Models - Gaussian Mixture Models (HMM-GMM), which means HMM models where the states are modeled by GMMs and Deep Learning which are HMM models where each state is modeled with DNN [4]. HMM-GMM have played an important role in designing conventional recognizers because they are easy to train and have low computational cost [4]. On the other hand, the deep learning approach was firstly used for speech recognition in the late eighties and early nineties. However, that attempt had great limitations. For instance, a network with more than two hidden layers was rarely used due to its computational cost and it neither showed great improvements with respect to GMMs [4]

Thanks to the advances in computational power, DNN re-started in the recent years and have shown very good results in different applications. In [7] and [8], results using different acoustic models in ASR systems with different acoustic conditions are reported. Both works show that DNN-based models outperform classical systems based on GMMs. Different topologies of networks have been proposed in order to improve the ASR performance. In [9] three topologies are compared: (1) Recurrent Neural Network, (2) Long Short Term Memory (LSTM) and (3) Gated Recurrent Unit (GRU). The authors used a total of 378 audio recordings from the TED talks in English. The dataset contains files for training, validation and test. Spectrograms were used to train the acoustic model. The best WER was achieved using LSTM (65.04%). GRU showed similar results (67,42% WER) in a lower training time; GRU only ran for 5 days and 5 hours while LSTM required slightly more than 7 days.

More complex architectures based on end-to-end systems were recently proposed. In [10], the authors compared different "very deep models". Convolutional LSTM with a residual connection (reConvLSTM) were also introduced in the same work. Convolutional LSTM layers basically replace multiplication operations among parameters and inputs by convolutions. Their architecture consists of 2 convolutional layers, followed by 4 ResConvLSTM, and finally an LSTM Network in Network block. A total of 80 filterbanks with their deltas were used as feature set. The Wall Street Journal (WSJ) English corpus [11] was used to train and test the network. This database contains 73 hours for training and 8 hours for test. The model proposed by Zhang et al. showed a WER of 10,53%, while previous studies were around 18% using the same corpus.

In the same line, the authors in [12] proposed an end-to-end system where its input is the raw speech signal. To do that, they used a convolutional filter learning based on rectangular bandpass filters. This technique is called SincNet. The authors proposed to connect SincNet to a recurrent encoder-decoder structure trained in an end-to-end manner using joint CTC-attention procedure. In this work, it was used WSJ [11] and TIMIT corpus [13] for training and testing their model. The authors compared their system with traditional end-to-end models operating on Mel-filter-banks. For TIMIT database, their technique did not have improvements in comparison to traditional hybrid DNN-HMM due to the small amount of available training data (less than 5 hours). On the other hand, when using WSJ, the proposed technique obtained a top-of-line WER of 4.5%, outperforming all the baselines. The previous best score was 5.9% WER, which means an absolute improvement of 1.2%.

Other kinds of techniques as speech enhancement, domain adaptation, and data augmentation have also been studied with DNNs. The authors in [14] proposed the problem-agnostic speech encoder (PASE), a novel architecture that combines a convolutional encoder followed by multiple neural networks, called workers, tasked to solve self-supervised problems. The aim of each worker is to generate features extracted from the original speech signal as MFCCs, log power spectrum, gammatone features, waveform speech signal, among others. The needed consensus across different tasks naturally imposed meaningful constraints to the encoder, contributing to discover general representations and to minimize the risk of learning superficial features. Self-supervised training was performed with a portion of 50 hours of the LibriSpeech dataset [7]; TIMIT [13], DIRHA [15] and CHiME-5 [16] were used for training and testing the ASR systems. In order to validate this technique, the authors trained a hybrid DNN-HMM speech recognizer using different acoustic features such as MFCC, filter bank, Gammatone, and MFCC Gammatone filter bank. The features extracted from PASE architecture significantly outperform the other feature sets, with a relative improvement of 9.5% in the clean scenario and of 17.7% in noisy conditions using TIMIT.

Data augmentation is another useful procedure in order to gain additional improvements. The authors in [17] introduced SpecAugment, a simple data augmentation method for speech recognition. The augmentation policy consists in warping the features, masking blocks of frequency channels, and masking blocks of time steps. Listening, Attention and Spell (LAS) network was trained and tested with this strategy using Librispeech [7] and Switchboard [18] corpus. An 80-dimensional filter bank with delta and delta-delta acceleration was used as input. The authors were able to obtain state-of-the-art results on the both in LibriSpeech 960h and Switchboard 300h tasks using an end-to-end LAS networks by augmenting the training, surpassing the performance of hybrid systems even without the aid of a language model.

An adaptation technique was proposed in [19]. The aim was to simultaneously model narrowband (sampled at 8kHz) and wideband (sampled at 16kHz) speech data in an ASR system using a domain mapping based on Generative Adversarial Network (GAN). The authors used a variation of GAN called cycleGAN. The advantage of this technique is that the aligned data is not required. CycleGANs based on LSTM, CNN and ResNet architecture were compared. The authors used a 50hr subset of Broadcast News data which is provided at 16kHz sampling rate (Wideband - WB) [20]. The WB subset was then downsampled at 8kHz (Narrowband - NB) and again upsampled at 16kHz to create corresponding upsampled narrowband (UNB). The authors trained one recognizer for each data set (WB, NB, and UNB) and each one was validated in all domains. The results show a degradation in WER when there is a mismatch among the domains. When the UNB subset is mapped to WB using ResNet GAN and validated on the WB-based recognizer, the system achieved an improvement on performance by 1.8% absolute WER.

1.2.2 Emotion recognition from speech & automatic customer satisfaction estimation

Many techniques have been studied to develop SER systems. In [21], the authors proposed to recognize different emotions included in the German database EMODB (happiness, boredom, neural, sadness, anger, and anxiety). The recordings were modelled with a set of 120 harmonic features along with their Δ and $\Delta\Delta$. Their minimum, maximum, mean, median, and standard deviation were also computed at an utterance level, producing 1800-dimensional feature vectors per sample. Speaker-independent (SI) multi-class classification was performed using SVMs and the authors reported average recall values of 92.00%, 71.48%, 87.46%, 91.42%, 98,29%, and 91.92% for the aforementioned emotions, respectively. For the multi-class classification scenario, the authors reported an average accuracy of 79.51%, which is a relatively high and optimistic accuracy because the validation strategy reported by the authors was a k-fold cross-validation, which does not guarantee unbiased results. The authors in [22] worked also with the EMODB corpus and modeled the emotions with temporal, spectral and cepstral features. Statistical functionals were computed per feature vector at an utterance level. The authors reported a maximum accuracy of 80% in the multi-class classification of the different emotions included in the dataset. Note that all of the experiments were speaker-dependent (SD), which leads to optimistic and biased results. Another relevant work in the topic of automatic emotion recognition was presented in [23]. The authors proposed a feature set that included frequency, energy/amplitude, and spectral parameters to model the speech signals. Mean, standard deviation, 20th, 50th and 80th percentiles, range between 20th and 80th percentile were the functionals computed per feature vector. The authors used a wide variate of datasets (TUM AVIC, GEMEP, EMODB, SING, FAU AIBO, and Vera-am-Mittag) to map information from the affective speech domain to the two-dimensional arousal and valence representation. An SVM classifier was trained following a leave-one-speaker-out cross-validation strategy to classify between high vs. low arousal and between positive vs. negative valence. The accuracy reported in the first scenario was 79.71% and 66.44% for the second one. Other approaches typically used to create SER systems are based on speaker representation models. For instance, the authors in [24] used i-vectors to classify emotions in two different datasets: (1) USC AudioVisual data [25] and (2) IEMOCAP [26], which are acted and spontaneous, respectively. The authors trained an SVM for the classification experiments and compared their approach w.r.t. the feature set proposed in the Interspeech 2010 Paralinguistics Challenge (I2010PC) [27]. According to their results, the system based on i-vectors yielded better performance (91.1% for USC and 71.3% for IEMOCAP) than I2010PC. It is important to highlight that these results were achieved on SD experiments and the hyper-parameter optimization procedure was not explained in detail, so it is not possible to know whether these results are optimistic and possibly biased.

More robust SER systems have been developed using deep learning techniques in the recent years. In [28], the authors compared 3 different network architectures: (1) Convolutional Neural Network (CNN); (2) Artificial Neural Network (ANN); and (3) Long Short Term Memory (LSTM). The IEMOCAP dataset was used to evaluate the approaches following an eight-fold-leave-two-speakers-out cross-validation scheme in all experiments. Log-Fourier transform -based filter-bank with coefficients distributed on the Mel scale were extracted. Because the authors performed a multi-class classification at a frame level, they assumed that frames belonging to a given utterance convey the same emotion as the parent utterance. Silence class was added using the labels generated by a voice activity detection system. The method was compared with prior works in the literature related to multi-class emotion classification that used the same database. Given that most works reported results at an utterance level, the posterior class probabilities computed for each frame in an utterance were averaged across all frames and an utterance-based label was selected based on the maximum average class probabilities. The authors reported that their system outperformed all the other methods in up to 2% of accuracy. In 2020, the authors in [29] explored possible dependencies between speaker recognition and emotion recognition topics. They first applied the transfer learning technique to transfer information from a ResNet-based pre-trained speaker-identity-based model to an emotion classification task. They also explored how the performance of a speaker recognition model is affected by different emotions. The authors evaluated their experiments on three different datasets: IEMOCAP, MSP-Podcast, and Crema-D. Two approaches were explored: x-vectors extraction and replacement of the speaker-discriminative output layer with an emotion classification layer and then fine-tune the hyper-parameters. A total of 23 MFCCs with 25ms frame-size and 10ms frame-shift were extracted. The model was validated on SI scenario. The best results were obtained with the transfer learning approach with accuracies of up to 70.3%, 58.46%, and 81.84 for IEMOCAP, MSP-Podcast, and Crema-D, respectively. In the same year, the authors in [30] proposed a one-dimensional CNN architecture to model emotions from speech. The input to the architecture was based on MFCCs, chromagram, Mel-scale spectrogram, Tonnetz representation, and spectral contrast features extracted from the audio files. The authors performed CS experiments with 4 classes of the IEMOCAP corpus and reported accuracies of 64.3%. The experiments with RAVDESS and EMODB were SD and the reported accuracies were 71.61% and 81.1%, respectively. Also in 2020, a novel technique was introduced in [31]. The authors proposed a method where key segments are selected based on a radial basis function network (RBFN) similarity measurement. The segments were grouped following the k-means algorithm as follows: The audios were divided into multiple chunks of 500ms and the RBFN similarity was computed. If consecutive segments did not exceed a given threshold, the number of clusters was increased by one. Thus, the number of clusters k dynamically changed. Once the segments were clustered, the nearest segment with respect to the centroid was selected for each cluster to generate a new sequence, which was then converted into a spectrogram. The spectrogram was passed through a CNN network (pre-trained ResNet101) to extract high-level features. These features were normalized and used as inputs to a deep bi-directional long short term memory (BiLSTM), which made the final decision about which emotions were present in a given recording. The authors considered the three "standard" databases IEMOCAP, EMODB, and RAVDESS and reported SI accuracies of 72.25%, 85,57%, and 77.02% for IEMOCAP, EMODB, and RAVDESS, respectively.

On the other hand, there are studies that investigated potential benefits of emotion analysis for estimating or modelling CS. The authors in [32] found a qualitative evidence that emotions identified from customers act as a proxies to estimate CS. They designed a system that interactively provides views of ongoing dialogues at different granularity to improve the quality assurance and CS evaluations. The authors considered 8 different emotions: happiness, assurance, agreement, courteousness, apology, unhappiness, disagreement, and no-emotions. These emotions are grouped into positive, negative and neutral sentiments. Real-life chat dialogues were collected forming a total of 188 conversations about mobile phone service providers. The conversations were manually annotated according to the aforementioned emotions. Each conversation was automatically split into turns (for customers and agents). To represent each emotion, the authors extracted simple NLP parameters like word dictionary-based counts, and also meta-content like the delay between two consecutive turns. To predict the emotions in each turn, they applied Conditional Random Fields (CRF). The model yielded accuracies of 62.4% for the 8-class emotion classification problem and 69.5% for the classification of the three different groups of emotions (positive, negative and neutral). The experiments only used the costumers' turns. The authors demonstrated the superiority of using their system over the use of manual quality assurance. A similar approach was proposed in [33]. Unlike the previous work, CS estimation was performed upon calls. The authors considered both acted and spontaneous call-center databases. The calls were annotated based on three emotion classes: positive, negative, and neutral. CS was predicted at a turn and call level. The CS estimation was performed following three steps. Firstly, the calls were split into turns by using a Voice Activity Detector (VAD). Secondly, emotions were predicted in each turn using a BiLSTM with prosody, lexical (using an ASR), and interactive features as input. Thirdly, CS was estimated by integrating turn-level emotion recognition through an unidirectional LSTM. The authors reported accuracies of 80.1% and 78.0% at the turn and call level, respectively. A less sophisticated approach is proposed in [34], where the authors released the Davero corpus, which contains phone-calls of a German call-center. The topics of the calls range from simple informative ones to complaint calls. The database contains about 93h of 1447 dialogues with 46610 turns. A total of 1,587 turns were annotated by psychology students, who considered a total of six emotions. The labels were clustered in terms of high and low dominance as well as positive vs. negative valence. To recognize the turn-level emotions, the authors extracted different acoustic features by using the Emotion and Affect Recognition toolkit (openEAR), and the classification was performed with an SVM. The most promising experiment presented by the authors was based on speaker-specific groups (i.e., speaker-group dependent) in which male and female speakers were considered separately. According to the results, the classification of positive vs. negative valence is about 82% of accuracy for male and about 70% for female.

1.3 Objectives

1.3.1 General Objective

To design, implement and evaluate an ASR system robust against non-controlled acoustic conditions and useful to automatically evaluate customer satisfaction.

1.3.2 Specific Objectives

1. To define at least two HMM-DNN-based architectures for ASR systems robust against non-controlled acoustic conditions.

2. To define at least two feature preprocessing techniques derived from domain adaptation, speech enhancement, and data augmentation to improve the noise robustness.

3. To implement and compare the previously defined techniques in order to select the most robust one against non-controlled acoustic conditions.

4. To evaluate the performance of the selected system according to the obtained WER and the automatic QoS evaluation.

1.4 Research Problem

The aim of this research work is to develop an efficient ASR system in non-controlled acoustic environments trying to maintain its performance in controlled conditions using a database collected at the Konecta Company. This research work defines, implements and evaluates different topologies and network architectures in controlled and non-controlled acoustic conditions in order to find the most accurate model. In addition, preprocessing techniques derived from data augmentation, speech enhancement, and domain adaptation are implemented to improve the ASR performance. Finally, in order to evaluate the utility of a DL-based ASR in a real application, automatic QoS evaluation performance is also analyzed by using the selected techniques on a QoS database provided by Konecta.

1.5 Contribution of the research work

In order to fulfill the objectives proposed in Section 1.3, this study is divided into 3 steps: (1) implementation of a speech recognizer robust against noise, (2) emotion classification & CS estimation from speech, and (3) multimodal CS analysis using speech and text features. The following subsections describe the contribution of each step.

1.5.1 Implementation of a speech recognizer robust against noise

With the aim of developing a hybrid ASR system robust against noise for call center application, a Complex Linear Coding (CLC) technique is considered to enhance the speech signals. This technique consists of applying a linear combination of complex coefficients to the complex spectrum of a speech signal. This master's thesis also compares different state-of-the-art hybrid ASR architectures based on TDNN, GRU, and LSTM. Finally, the performance of the selected system is evaluated in automatic QoS evaluation.

1.5.2 Emotion classification and customer satisfaction analysis from speech

With the aim to address the problem of automatic speech emotion classification and also the problem of modeling CS from speech recordings collected under non-controlled conditions, this research study introduces the use of phonation, articulation and prosody features extracted with the DisVoice framework [35]. The feature sets are also used in different emotional speech corpora widely used in the related literature. The results obtained with the introduced feature sets are compared with respect to three different approaches: two speaker models namely, i-vectors and x-vectors, and the I2010PC[1] feature set [27]. The results show that the proposed approach is competitive when considering the "standard" emotional speech databases and it is the better one when considering the recordings with the opinions of customers of the call center, which were collected under non-controlled acoustic conditions. In addition, the following reasons support the fact that the proposed feature set may be more suitable for industrial applications:

[1] I2010PC and openSMILE are used indifferently to refer to the same feature set

1. It considers a smaller number of features (i.e., 488 for articulation) vs. 1582 from openSMILE.
2. There are no restrictions for its commercial use.
3. It was the best feature set along the experiments about CS analysis.
4. This feature set shows more balanced results in terms of sensitivity and specificity compared to the openSMILE feature set.
5. The introduced feature set can be used in real-world applications where the CS needs to be evaluated solely based on acoustic information.

1.5.3 Multimodal CS analysis using speech and text features

With the aim of improving the automatic estimation of CS, this research work performes a multimodal analysis based on four different feature sets: (1) speaker embeddings based on x-vectors, (2) acoustic features based on openSMILE, (3) articulation features based on onset/offset transitions, and (4) text features based on word2vec using text transcriptions generated by an ASR system. Two different fusion schemes are explored in order to robust recognize CS: Early Fusion (EF) and Gated Multimodal Unit (GMU). It is hypothesized that GMU is more suitable to combine the features sets, since it can control the importance of each set according to its degree of contribution to encode the internal pattern (i.e, CS in this study). The fusion schemes are evaluated through SVM and DNN classifiers on real-world database collected from a call center.

1.6 Structure of the research work

This master's thesis is divided into six chapters:

Chapter 1: includes the motivation, the state-of-the-art, the objectives, the problem statement and the contributions of this master's thesis.

Chapter 2: introduces the basic concepts of an ASR system, which constitutes the main topic of this research work. In addition, the most commonly used architectures in hybrid recognizers are discussed.

Chapter 3: describes the databases considered in this work. This chapter is divided into two sections: data for ASR implementation and data for emotion & CS evaluation. The sections describe in detail the datasets used for each system.

Chapter 4: explains the methodology followed for each system. The first section describes the process for developing an ASR system robust to noise. The second section details the methodology designed for emotion classification and CS evaluation from speech. The third section describes the implementation of multimodal analysis for CS evaluation.

Chapter 5: presents the results obtained for each experiment and some discussion of such results.

Chapter 6: contains the conclusions and future work potentially derived from each experiment.

Chapter 2

Theoretical background

This chapter introduces the basic concepts of an ASR system. It then discusses some architectures of the acoustic model in more detail. Finally, it briefly describes how speech recognition systems can be integrated for CS analysis in call centers.

2.1 Automatic Speech Recognition System

The performance of an ASR should be independent of certain conditions such as microphone, speaker accent, and acoustic environment. The general procedures of an ASR system are shown in Figure 2.1. The system essentially consists of an acoustic processor called “Feature Analysis” and a decoder called “Pattern Classification”. The acoustic processor converts the speech signal $s(t)$ into a feature set $\mathbf{o}_t$, which is a compact representation of $s(t)$. The pattern classifier decodes the sequence of features, $\mathbf{o}_t$, into a word sequence, $\hat{\mathbf{W}}$, which is the most likely sequence given $\mathbf{o}_t$. The decoder uses an Acoustic Model (AM). The AM is represented by the posteriors obtained from a Hidden Markov Model (HMM). Finally, a dictionary is used to compute the matched probability between the sequence of features and the most likely word sequences. Additionally, an N-gram Language Model (LM) is used to compute the word sequence probability according to a given grammatical context.

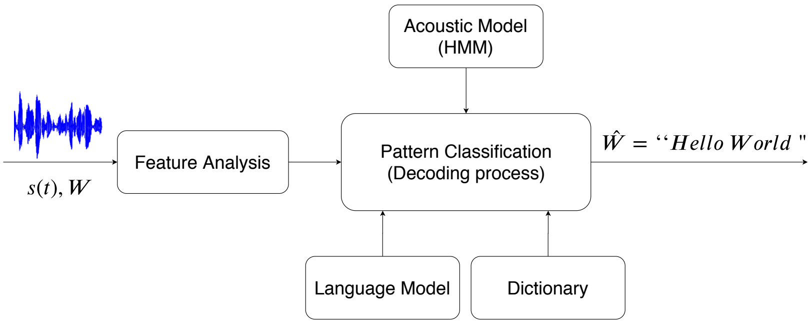


Figure 2.1. General structure of an ASR system. Figure adapted from [2].

2.2 Feature analysis

Features are a set of numbers that represent a speech signal. This set is named feature vector. There is no standard feature set, however, the most popular in ASR is Mel Frequency Cepstal Coefficients (MFCC) [36]. Before the feature extraction, it is needed to preprocess the speech signal. Thus the first procedure is to convert it into a digital signal applying Nyquist sampling theorem and then quantized according to the desired quality. Then, MFCCs are computed as shown in Figure 2.2. Given that the speech signal's characteristics change slowly (varying every 50-100ms), when analyzing them into short segments, it is possible to assume a stationary behavior and model them with a linear and time-invariant system. Therefore, the signal is divided into frames of N samples which are spaced M samples apart. N and M typically correspond to frames of duration 15-40ms and frameshifts of 10ms [2]. The border of these segments has discontinui-ties which do not correspond to a real-wold signal properties. The speech segments are thus windowed using smooth functions (Hamming, Gaussian, Blackman, others) to reduce the segmentation effects. The windowed signal is then transformed to the frequency domain using Fast Fourier Transform (FFT) and filtered on the mel-scale using triangular filters. These filters aim to model the non-linear human ear perception with better discrimination at lower frequencies. The log() function is applied for each power spectrum resulting in each filter bank. Finally, a Discrete Cosine Transform (DCT) is computed to obtain the MFCCs.

In addition, the recognition performance can improve when the dynamic

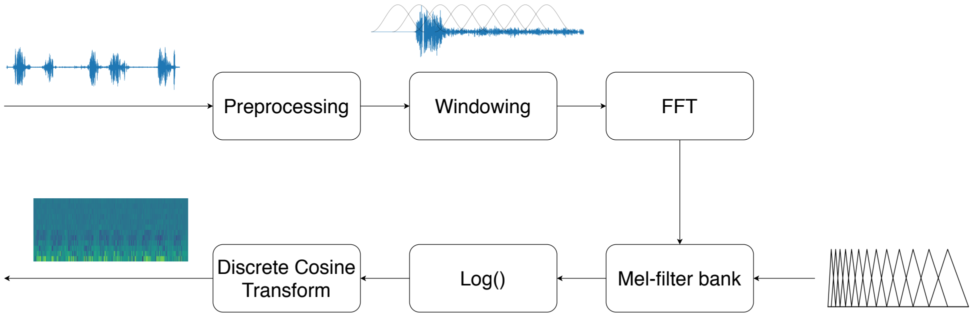


Figure 2.2. Procedure for extracting MFCCs.

of the power spectrum is added to the feature space, i.e, delta (differential) and delta-delta (acceleration). This is important in pronunciation, since articulations, like stop closures and releases, can be recognized by the formant transitions [37]. The first derivative is thus computed as:

$$\mathbf{d}_t = \frac{\mathbf{o}_{t+1} - \mathbf{o}_{t-1}}{2} \tag{2.1}$$

where $\mathbf{o_{t+1}}$ is the feature vector observed at time $t+1$ and $\mathbf{o_{t+1}}$ is the feature vector observed at time $t-1$. The second derivative is then computed also using the Equation 2.1 but replacing $\mathbf{o}_t$ by the first derivative $\mathbf{d_t}$.

2.3 Language Model (LM)

This is used to find words. It defines which word could be the following given the previous words (context). The LM has the task of assigning a probability for a sequence of words, $\boldsymbol{W}$, to be seen on a given context:

$$P(\boldsymbol{W}) = P(w_1, w_2, ..., w_M) \tag{2.2}$$

where w_n corresponds to the $n-$th word of the sequence $\boldsymbol{W}$.

The majority of the LMs are N-gram models which contain statistics of N-word sequences. If it is assumed that a likelihood of a word only depends on the $N-1$ last occurrences of words, the base of an $N-$gram language model is obtained. Thus, Equation 2.2 can be rewritten as:

$$P(\boldsymbol{W}) = \prod_{n=1}^{M} P(w_n | w_{n-1}, w_{n-2}, ..., w_{n-N+1}) \tag{2.3}$$

2.4 Dictionary

It contains all linguistic units that should be recognized. For the case of a phonetic dictionary, every word is mapped to its corresponding phone sequence. For instance, if a Spanish ASR system is being designed and it is desired to recognize the word “camisa”, one entry of the dictionary would be: “camisa: \k \a \m \i \s \a”, where each character corresponds to a phoneme.

2.5 Acoustic Model

An AM essentially represents the relationship between the speech signal and the phones or units that build the speech communication. Due to aforementioned advantages in Section 1.1, the models based on DNN are playing an important role on current speech recognizers. However, these models are static, thus they are not useful to model random time domain sequences. On the other hand, HMMs are able to model random sequence in time domain, but they are not useful to represent acoustic distribution of speech signals by themselves. Therefore, combining DNN and HMM, it is possible to obtain an appropriate speech signal representation. This combination is called HMM-DNN.

2.5.1 Deep Neural Networks

There is a number of architectures based on deep learning, each having its advantages and disadvantages depending on the application. Many architectures of the acoustic model of a hybrid speech recognizer have been explored, including: feed-forward Neural Network (DNN), (2) Convolutional Neural Network (CNN), Recurrent Networks (RNN, LSMT and GRUs), Time-delay Neural Network (TDNN), and others. On the one hand, although DNN-based models have already fallen behind in the state-of-the-art, it was considered in this work in order to compare the evolution of performance with respect to the other models. On the other hand, CNN-based models are shown high performance in speech recognition applications, they were not considered in this work since these architectures take advantage of the spatio-temporal correlation of the features. However, this work uses MFCCs, which are poorly correlated features.

DNN: It is a conventional MLP [38] that includes at least 2 hidden layers [4]. Figure 2.3 shows an example of DNN which consists of an input layer (the first one), three hidden layers and an output layer (the last one). Each circle at the hidden and output layers represents a neuron. This unit (neuron) is basically formed with weights associated to each input (interconnection line), a bias (represented by the circles with the number 1 inside) and an activation function. The role of each element in a neuron is: (1) The weights measure the degree of correlation between the activity level of the neuron (output) and the inputs that they connect [39]. It gives information about how much influence an input has in the outcome of the neuron. (2) The bias is a constant value independent of the previous neurons and produces an offset in the activation function. (3) The activation function controls the outcome of the neuron. It determines whether the neuron should be activated or not depending on its inputs, its weights and its bias. In some cases, this function also helps to normalize the output of the neuron between -1 and 1 or between 0 and 1. A neuron representation is depicted in Figure 2.4.

A neuron activation can be therefore calculated as:

$$v = f(z) = f(w_1 x_1 + w_2 x_2 + ... + w_N x_N + b) \tag{2.4}$$

where w_i is the weight associated to the input x_i, b is the bias and N is the number of inputs of the neuron.

Since, a DNN is formed by several layers and each layer is composed of many neurons, the Equation 2.4 can be extended to a DNN as a matrix representation such as:

$$\mathbf{v}^l = f(\mathbf{z}^l) = f(\mathbf{W}^l \mathbf{v}^{l-1} + \mathbf{b}^l), \quad \text{for} \quad 0 < l < L \tag{2.5}$$

where $\mathbf{z}^l = \mathbf{W}^l \mathbf{v}^{l-1} + \mathbf{b}^l \in \mathbb{R}^{N_l \times 1}$, $\mathbf{v}^l \in \mathbb{R}^{N_l \times 1}$, $\mathbf{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$, $\mathbf{b}^l \in \mathbb{R}^{N_l \times 1}$ and $N_l \in \mathbb{R}$ are, respectively, excitation vector, activation vector, weight matrix, bias vector and the number of neurons in each layer l. In this case, each row of the matrix $\mathbf{W}^l$ accordingly contains the weights of each neuron belonging to the layer l. In the same way, the bias vector $\mathbf{b}^l$ includes in each element the bias of each neuron in layer l. Due to the layer 0 represents the input layer, then $\mathbf{v}^0 = \mathbf{o} \in \mathbb{R}^{N_0 \times 1}$; being $\mathbf{o}$ the N_0-dimensional feature vector. The output vector $\mathbf{v}^L$ contains the posterior probabilities that the feature vector $\mathbf{o}$ belongs to each of the linguistic units (like a phoneme, or part of it).

RNN: The basic concept of an RNN was introduced in [40]. This model

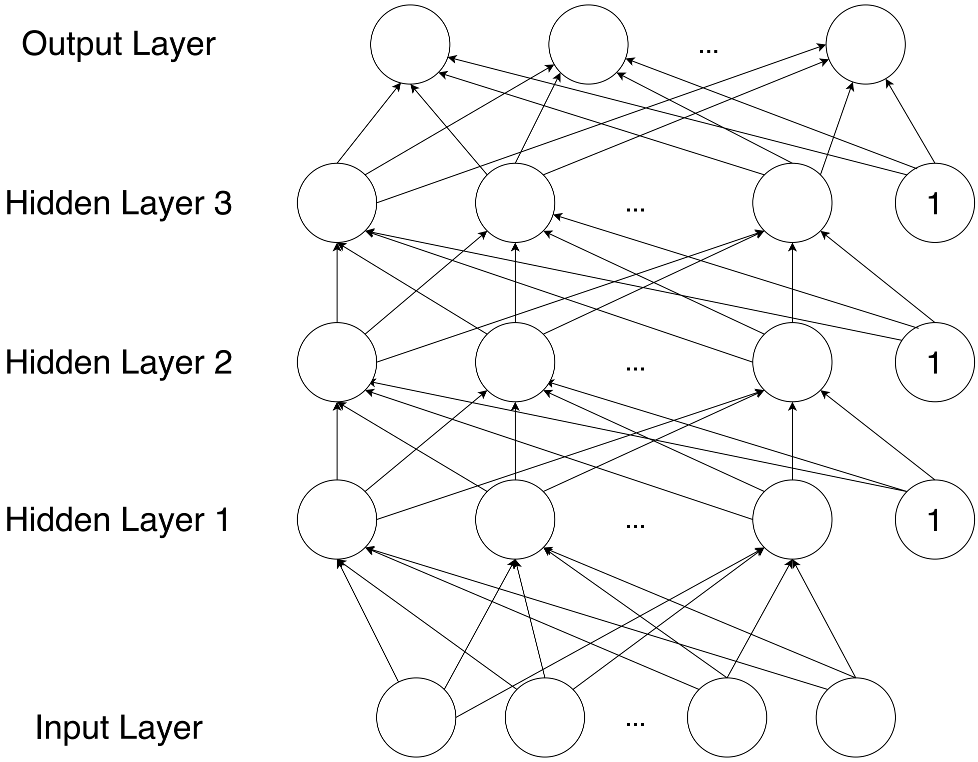


Figure 2.3. DNN structure. Figure adapted from [4].

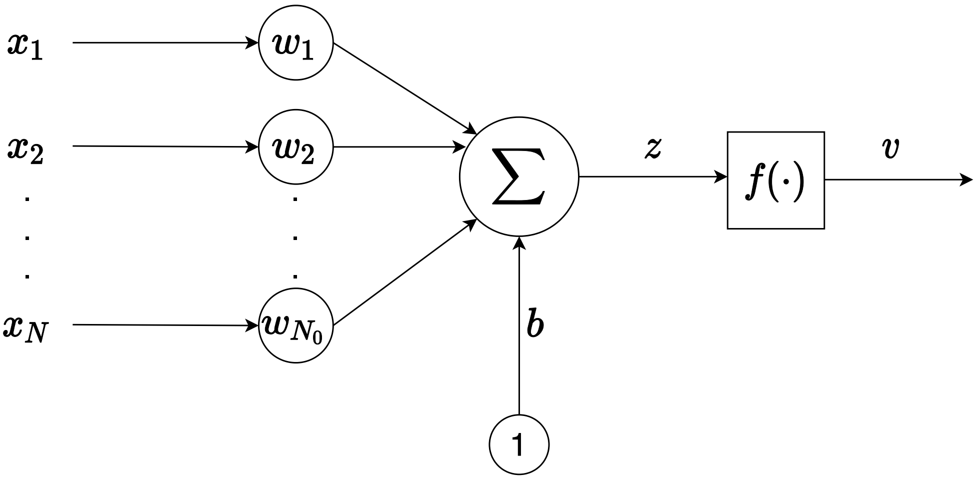


Figure 2.4. Graphical representation of a neuron.

operates based on the inputs and also on internal states (hidden state) that encode past information in a temporal sequence [4]. While GMM and DNN are static systems, RNN is dynamic. The hidden state in the RNN allows to learn representation of dependencies over a time span. For an one-layer RNN and each time step t, the mathematical formulation of an RNN is similar to a one-layer DNN but including the contribution of the internal state, as follows:

$$\mathbf{h}_t = f(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{2.6}$$

where $\mathbf{x}_t$ is the K-dimensional input vector, $\mathbf{W}_{xh}$ is the $N \times K$ matrix of weights connecting the K inputs to the N hidden units, $\mathbf{W}_{hh}$ is the $N \times N$ matrix of weights connecting the N hidden units from the time $t-1$ to time t, and $\mathbf{b}_h$ is the N-dimensional bias vector. The internal state h_t is then used as input to the fully-connected layer (v_t) to compute the posterior probability of the linguistic units as follows:

$$\mathbf{v}_t = f(\mathbf{W}_{hv}\mathbf{h}_t + \mathbf{b}_v) \tag{2.7}$$

where $\mathbf{W}_{hv}$ is the $S \times N$ matrix of weights connecting the N hidden units to the S emission probabilities of the linguistic units, $\mathbf{h}_t$ is the hidden state at the time t and $\mathbf{b}_v$ is the bias vector.

Figure 2.5 shows the structure of an RNN containing multiple recurrent layers $(\mathbf{h}^l)$ and a feed forward neural network $(\mathbf{v}^L)$ as the output. In this case, the recurrent layer $\mathbf{h}^l$ is fed by the hidden state of the previous recurrent layer $\mathbf{h}^{L-1}$. The output layer $\mathbf{v}^L$ computes the posterior probabilities per linguistic unit.

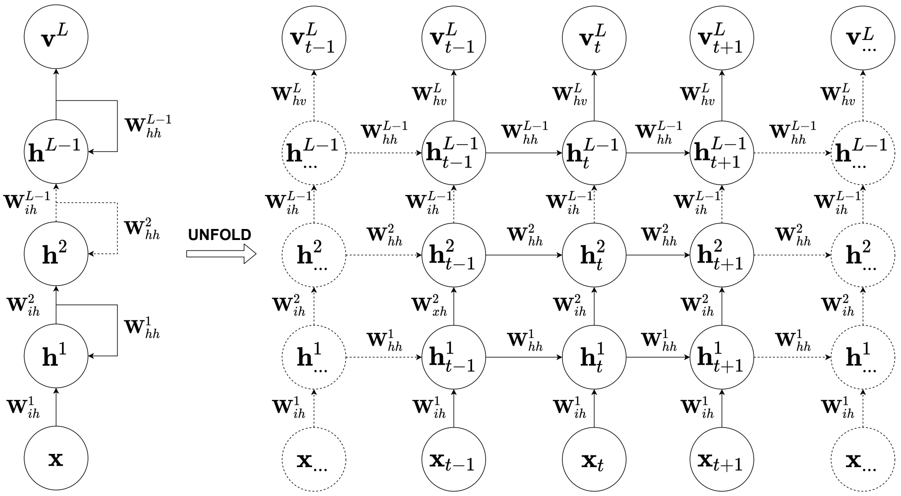


Figure 2.5. Unfold structure of an RNN.

LSTM: As mentioned above, the main advantage of the RNNs is their ability to encode past information in a temporal sequence. However, these architectures have the disadvantage of rather limited contextual range. That is, the influence of an input on the network decays exponentially over time as new inputs are fed into the network. This problem is also known in the literature as vanishing gradients. Thus, the authors in [41] the Long Short-Term Memory (LSTM) to solve this problem.

This architecture consists of a set of special units recurrently connected, called memory blocks. The LSTM blocks contain memory cells that store temporal state of the network as well as multiplicative gates to control the flow of information over time. Each memory block includes an input gate, a forget gate and an output gate. The input gate controls the flow of information from the input into the cell memory. The forget gate weights the amount of information allowed from the cell's internal state before feeding it into the cell. The output gate controls the flow of information from the block into the rest of the network. For an one-layer LSTM and each time point t, the mathematical formulation is as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{2.8}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f) \tag{2.9}$$

$$\mathbf{g}_t = \tanh(\mathbf{W}_{xg}\mathbf{x}_t + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_g) \tag{2.10}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o) \tag{2.11}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_{t-1} \tag{2.12}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{2.13}$$

where $\mathbf{x}_t$ is the $K-$dimensional input vector, $\mathbf{h}_t$ is the N-dimensional hidden state, and $\mathbf{c}_t$ is the N-dimensional cell state. $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{g}_t$, and $\mathbf{o}_t$ are the input, forget, cell, and output gates, respectively. The weight matrices ($\mathbf{W}$) and the bias vectors ($\mathbf{b}$) connect the input-vector/hidden-units to their respective gates/cells. σ is the sigmoid function, tanh is the hyperbolic tangent function,

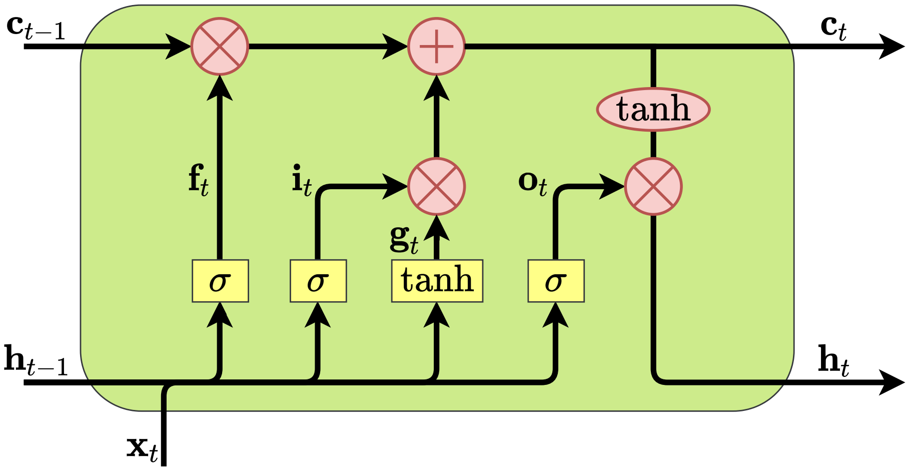


Figure 2.6. Structure of an LSTM unit.

and $\odot$ is the Hadamard product. Figure 2.6 shows the structure of an LSTM cell.

GRU: This model is also based on gating mechanism, and introduced in [42] using a machine translation task. Similar to the LSTM, the GRU has multiplicative gates that control the flow of information with the difference that it does not have memory cells. The flow of information is controlled through two gates: the update gate and the reset gate. On the one hand, the update gate controls how much new information updates its internal state based on a linear interpolation between the previous state and the candidate state. On the other hand, the reset gate decides whether the previous state is important or not. Thus, the mathematical formulation of a one-layer GRU unit is as follows:

$$\mathbf{r}_t = \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{2.14}$$

$$\mathbf{z}_t = \sigma(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{2.15}$$

$$\mathbf{n}_t = \tanh(\mathbf{W}_{xn}\mathbf{x}_t + \mathbf{r}_t \odot (\mathbf{W}_{hn}\mathbf{h}_{t-1}) + \mathbf{b}_n) \tag{2.16}$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{z}_t) \odot \mathbf{n}_t + \mathbf{z}_t \odot \mathbf{h}_{t-1} \tag{2.17}$$

where $\mathbf{x}_t$ is the K-dimensional input vector and $\mathbf{h}_t$ is the N-dimensional hidden state. $\mathbf{r}_t$, $\mathbf{z}_t$, and $\mathbf{n}_t$ are the reset, update,, and new candidate gates. The weight matrices ($\mathbf{W}$) and the bias vectors ($\mathbf{b}$) connect the input-vector/hidden-units to their respective gates. σ is the sigmoid function, tanh is the hyperbolic tangent function, and $\odot$ is the Hadamard product. Figure 2.7 shows the structure of a GRU cell.

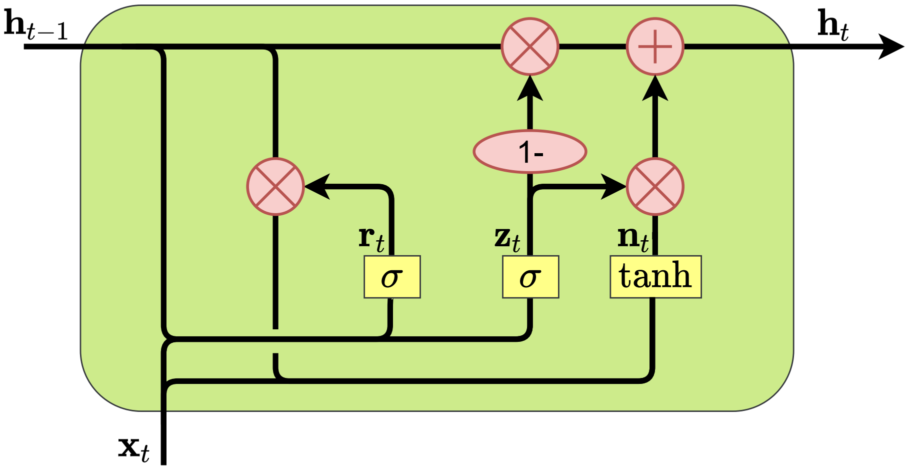


Figure 2.7. Structure of a GRU unit.

TDNN: Although recurrence-based models have been shown to model temporal dependencies effectively, their training time is higher than, for instance, DNNs due to the sequential nature of the training algorithm [43]. Waibel in [44] designed the TDNN, an alternative for modeling long term temporal dependencies with training times comparable to a standard DNN. The main difference between a DNN and a TDNN is the way in which the time dependency is transformed. On the one hand, when modeling long temporal context in a standard DNN, the initial layer performs a linear transformation to the entire context. On the other hand, the initial layer of a TDNN transforms the input from a narrow temporal context and the deeper layers process the hidden activations from a wider temporal context. In addition to the hyperparameters defined in the DNN network, the TDNN requires defining the input context for each of its layers in order to compute the activation function at each time instant. Figure 2.8 shows an example of the temporal dependencies of each layer for different time steps in a TDNN. The figure shows two possible TDNN configurations: without sub-sampling (black lines) and with sub-sampling (red lines). In the fist case, the hidden activations are computed at all time steps. In the second case, gaps between frames are allowed, i.e., instead of splicing contiguous windows, the network splices frames separated by at least one time step. The model takes advantage of possible correlations between contiguous windows. In this way, it is possible to omit some frames in the context in order to reduce the computational cost.

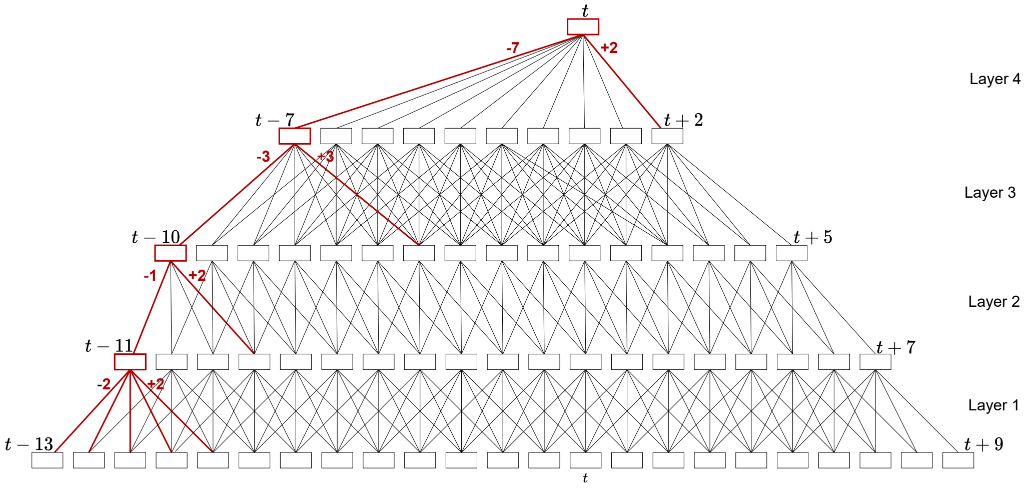


Figure 2.8. Computation in TNN with sub-sampling (red) and without sub-sampling (black). Figure adapted from [43]

2.5.2 Markov chain

Before explaining what a Markov sequence is, it is necessary to first explain the Markov chain. It is a special case of a Markov sequence where the likelihood of a given event only depends on the immediately previous state [4]. This chain, $\boldsymbol{q_1^T} = q_1, q_2, ..., q_T$, can be defined by its transition probabilities a_{ij}:

$$P(q_t = s^{(j)} | q_{t-1} = s^{(i)}) = a_{ij}(t), \quad \text{where} \;\; a_{ij} \geq 0 \; \forall i, j; \;\; \text{y} \; \sum_{j=i}^{N} a_{ij} = 1 \quad (2.18)$$

and by its initial state-distribution probabilities $\hat{\pi}(s^{(i)}) = P(q_0 = s^{(i)})$. Thus, the possible values that a random variable q_t of a Markov chain can get are: $q_t \in \{s^{(j)}, j = 1, 2, ..., N\}$. Additionally, if the transition probabilities are time independent, it is obtained a homogeneous Markov chain. Figure 2.9 depicts a 3-state homogeneous Markov chain being a_{ij} the transition probabilities from the state $s^{(i)}$ to the state $s^{(j)}$.

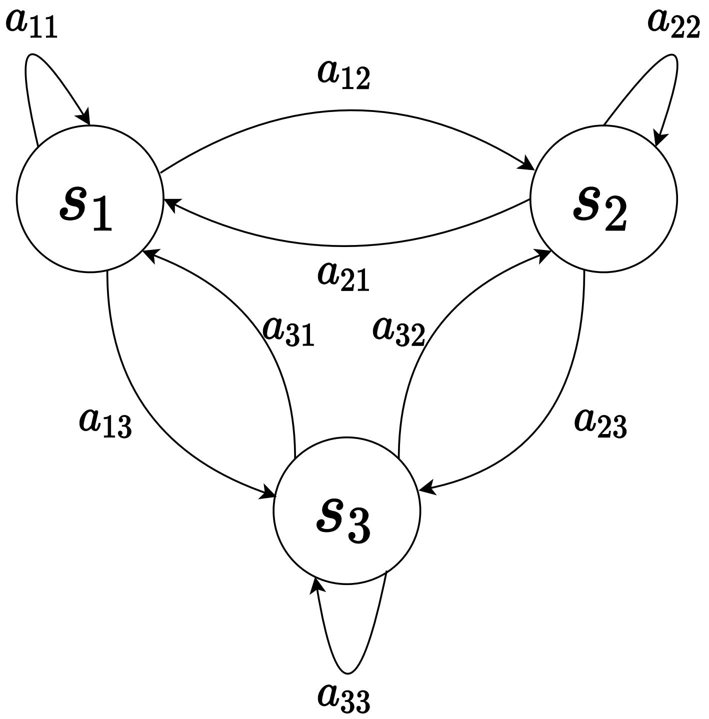


Figure 2.9. Markov Chain.

Therefore, the state-occupation probability, $p_i = P(q_t = s^{(i)})$, can be recursively computed as:

$$p_i(t+1) = \sum_{j=1}^{N} a_{ji} p_j(t), \quad \forall j. \tag{2.19}$$

where $p_i(0) = \hat{\pi}(s^{(i)})$.

2.5.3 Hidden Markov sequence

As shown in the previous definition, a Markov chain has a one-to-one correspondence between the state and the output of the sequence. Therefore, a

Markov chain is an observable Markov sequence because there is no randomness in the output in a given state. For instance, if the output of each state of a Markov chain is defined as follows:

$$s_1 = 2, s_2 = 3, s_3 = 6$$

and an output sequence is observed as: $q_1^5 = \{2, 6, 3, 2, 2\}$, it can be claimed that the state sequence follows by the model was $\{s_1, s_3, s_2, s_1, s_1\}$. This characteristic makes the Markov chain restrictive to model many real-world information sources, such as speech feature sequences. If the speech communication is considered as a sequence of phonemes, this process is not possible to model with a Markov chain because the acoustic waveform of each phoneme (state) is a random variable and not a discrete value. When the discrete state value is replaced by a random variable, the Markov chain is then generalized to Markov sequence [4]. If the random variable distribution is overlapped among the states, the new sequence is called a hidden Markov sequence.

The hidden Markov sequence then is defined by:

- Transition probabilities, a_{ij} of an homogeneous Markov chain and the initial state-occupation probabilities $\hat{\pi}_i$.
- Observation probabilities, $b_i = P(\boldsymbol{o_t}|s^{(i)})$ being $\boldsymbol{o_t}$ a random vector observed at time t and $s^{(i)}$ the state i which embeds the probability distribution function (PDF) b_i.

The most common PDF for representing speech features are GMM and DNN. Thus, when the hidden Markov model embeds GMMs in its states, the model is called HMM-GMM (Figure 2.10) but when it uses a DNN for modeling the speech features, it is called HMM-DNN (Figure 2.11). As shown in the figures, both of them use an HMM to represent transition probabilities $a_{i,j}$ in the time series (acoustic feature sequence). However, the only difference is that the first one computes the emission probability $b_i(\mathbf{o_t})$ using a GMM and the second one uses a DNN. Note that the DNN can also be replaced by more complex deep learning architectures such as those mentioned above: RNN, GRU, LSTM, and TDNN.

Each state of the HMM describes a simple unit of a speech signal (a complete phoneme or part of it). As shown in the Figure 2.11, when joining those units like a sequence, more complex models are produced such as words

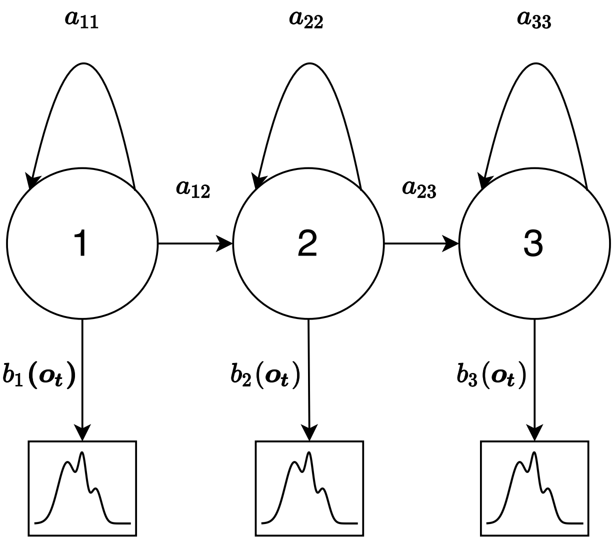


Figure 2.10. Basic structure of an HMM-GMM with 3 states.

or even sentences. Consequently, when performing whatever combination of states, it is possible to generate all words belonging to the language to be recognized.

2.5.4 Deep Neural Network embedded into a Hidden Markov Model

As explained in the last section, an HMM-DNN is a Marchov chain that embeds in each state a random variable whose PDF is represented by a DNN. In addition, it is possible to represent complex models like words just building an HMM-DNN with several states, depending on how many linguistic units (as phonemes) a word has. For instance, the Spanish word “casa” would have fewer states than the word “camisa”, because the first one has four phonemes while the other one has 6 phonemes. In the same way, it is possible to build even more complex models like sentences, joining word models based on HMM-DNN.

The conditional likelihood $P(\mathbf{o}_1^T|w)$ that the observation sequence $\mathbf{o}_1^T$ is generated by the word w is defined as:

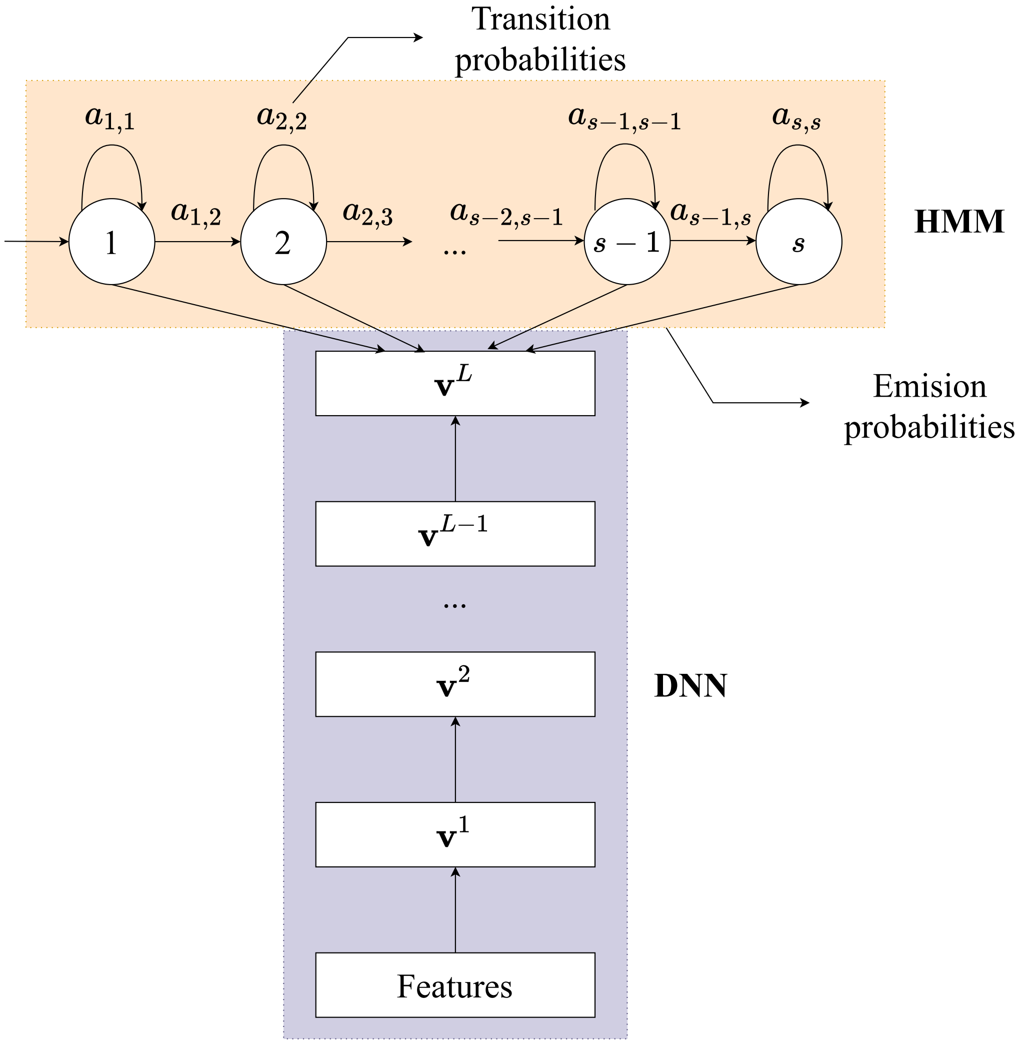


Figure 2.11. Basic structure of an HMM-DNN with s states. Image taken from Yu 2015.

$$\begin{aligned} P(\mathbf{o}_1^T|w) &= P(\mathbf{o}_1^T|\mathbf{q}_1^T, w)P(\mathbf{q}_1^T|w) \\ &\approx \max\{\hat{\pi}(q_0)\prod_{t=1}^{T} a_{q_{t-1}q_t}\prod_{t=1}^{T} p(q_t|\mathbf{o}_t)/p(q_t)\} \end{aligned}$$

where $\mathbf{q}_1^T$ represents the all possible state sequence with T frames in the model of the word w, $a_{q_{t-1}q_t}$ represents the transition probabilities from the state q_{t-1} seen at time $t-1$ to the state q_t seen at time t, $\hat{\pi}(q_0)$ represents the initial-occupation probability in the state q_0 seen at time 0, $p(q_t|\mathbf{o}_t)$ is the posterior probability that is computed from the DNN, and $p(q_t)$ represents prior probability of each state seen at time t. This procedure is computed over all words (or sentences) in order to search the most probable one. Since, a sentence $\boldsymbol{w}$ is a word sequence (which is finally a large HMM), this formulation can be generalized to a sentence by evaluating the conditional likelihood $P(\mathbf{o}_1^T|\boldsymbol{w})$ over whole sentence $\boldsymbol{w}$ instead of an isolated word w.

2.6 Automatic Speech Recognition system for emotion recognition and customer satisfaction evaluation

Automatic emotion recognition systems were introduced several years ago and have evolved a lot since then. It has been shown that those systems are suitable to help call-center managers in monitoring and optimizing the QoS provided by their agents [45]. These systems can potentially detect the emotional state of agents and/or customers and hence provide a QoS index.

The automatic QoS analysis can be rated by evaluating the customers satisfaction (CS). There are two main approaches to evaluate it: acoustic analysis and text analysis. In the fist case, the system detects abnormal changes on speech signals such as: poorly-articulated speech, increase in speech rate, increase in voice volume and others. In the second case, the evaluation is based on the linguistic content of the speech. For instance, the system searches for keywords and sentences reflecting satisfaction/dissatisfaction. However, the text-based approach requires including an ASR system that can convert the speech signal into a text transcription efficiently. The recognizer must be designed in such a way that its dependency to the acoustic conditions is minimal. If the recognizer performs poorly, the automatic satisfaction evaluation system will also perform poorly.

A more appropriate approach consists of fusing both the acoustic analysis and text analysis because both representations provide complementary information. This technique is also known as multimodal analysis due to the fusion of different modes (i.e., acoustic and text modes). The two most basic schemes of combination of characteristics are early fusion (EF) and late fusion (LF). On the one hand, EF has the disadvantage of combining features of different nature. In addition, it assumes that each modality has the same importance (weight) for classification purposes. One the other hand, disadvantage of LF is the high training cost due to the need to train one model for each modality. Additionally, it requires an additional training stage to optimize the classifier that makes the final decision. For this reason, the Gated Multimodal Units (GMU) were proposed in [46] to more adequately fuse features. A GMU is DL-based model that combines ideas from both EF and LF. A GMU learns to decide how much each modality influences the activation of the unit using multiplicative gates. Since information sources (modalities) usually do not share statistical properties, the GMU-based approach is suitable because it learns intermediate representations through linear combinations of the different modalities.

Chapter 3

Databases

This chapter describes the databases used for ASR implementation and emotion & CS evaluations respectively.

3.1 Data for ASR system implementation

To evaluate the ASR systems, call recordings of the Konecta company are used. Two data augmentation schemes are also considered: (1) using additive noise taken from the Demand Noise Dataset (DND); and (2) using signal speed perturbation.

3.1.1 Konecta calls

This corpus contains recordings of conversations between customers and agents of a contact-center of the Konecta Group (Medellín, Colombia). The customers were informed that their speech was going to be recorded. Due to the nature of the service, it is assumed that the speakers in these recordings are all of legal age. The database consists of $478,6$ hours of audio with a sampling frequency of 8kHz and a 16 bits resolution. Experts in QoS annotated the recordings in the contact center. Each audio has its transliteration, the customer's gender, and its level of noise, which was perceptually labeled by the experts as low, medium and high. Since the recordings were captured in non-controlled acoustic environments, this database is useful to evaluate the robustness of ASR systems against noisy conditions. Table 3.1 shows the demographic information of this database.

Table 3.1. Demographic description of the Konecta calls database. **LN:** Low level of noise. **MN:** Medium level of noise. **HN:** High level of noise. **Male:** Number of male recordings. **Female:** Number of female recordings
.

Label of noise	# of speakers	Gender distribution		Hours	
		Male	Female	Training	Test
LN	18938	19459	27313	321,0	30,3
MN	6615	7180	8191	101,4	15,9
HN	633	636	666	8,7	1,2

3.1.2 Demand Noise Dataset (DND)

The DND corpus [47] contains a variety of noise signals collected in real-world acoustic environments. The database considers two scenarios, namely "inside" environments and "open-air" environments. The inside recordings are divided into Domestic, Office, Public, and Transportation; while the open-air recordings are classified as Street and Nature. All recordings are captured with a 16-channel array of microphones at a sampling frequency of 48kHz. Thus, each environment noise recording is actually a set of 16 mono sound files.

3.1.3 Data augmentation based on additive noise

Clean recordings (LN) of the Konecta calls corpus are augmented by adding noise signals of the DND corpus. The noisy samples are created by randomly taking two different noises from the DND corpus associated with different Signal-to-Noise Ratio (SNR) levels: $-5, 0, 5, 10, 20,$ and $40\,\mathrm{dB}$. To achieve the selected SNR level, the noise is scaled by a factor α, which is expressed as:

$$\alpha = \sqrt{\frac{P_{s(t)}}{SNR \cdot P_{n(t)}}} \tag{3.1}$$

where $P_{s(t)}$, $P_{n(t)}$, and SNR are speech signal power, noise signal power, and SNR computed in linear scale, respectively.

Training and test sets are augmented separately and used to train and evaluate the denoiser described in Section 4.1.1. The data augmentation algorithm is depicted in Figure 3.1.

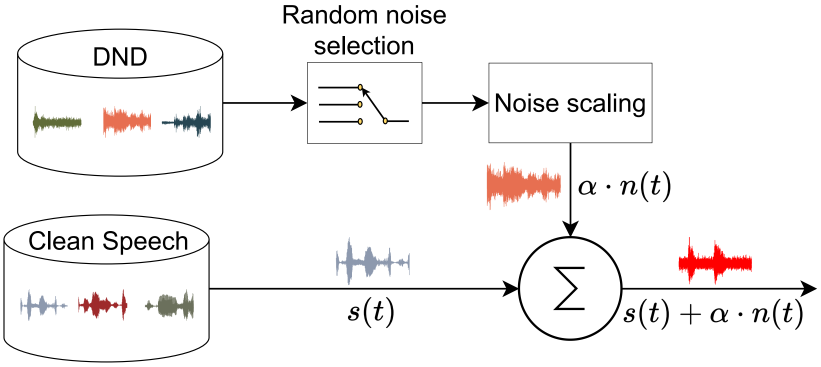


Figure 3.1. Data augmentation process.

3.1.4 Data augmentation based on speed perturbation

This perturbation applies a time warping by a factor β. Given a speech signal $x(t)$, the time-warped signal is $x(\beta t)$. The speed perturbation alters the signal duration as well as the number of frames in the utterance. The training sets (LN, MN, and HN) of the Konecta calls corpus are augmented with speed factors of 0.9, 1.0 and 1.1. Thus, three versions of the original recording are generated by applying the speed perturbation. This scheme is implemented using the speed function of the *Sox* toolkit[1]. Details of this data augmentation technique can be found in [48].

3.2 Data for emotion recognition & customer satisfaction evaluation

This master's thesis considers three speech emotional databases commonly used in the literature of speech emotion recognition (SER): (1) IEMOCAP, (2) RAVDESS, and (3) EMODB. Each corpus contains audio recordings with emotional content. They constitute the standard databases for the training and evaluation of SER models. Besides these well-known corpora, here the Konecta voicemails database was introduced. This dataset was created with

[1] The toolkit is available on: http://sox.sourceforge.net

audio recordings of a Konecta Group. Unlike Konecta calls, this corpus is based on voicemails, which means that it is formed with recordings of messages that customers leave after receiving assistance from a call center agent. It is used to evaluate the proposed approach in real-world acoustic conditions. All datasets are down-sampled to 8kHz. Further details of each corpus are presented below.

3.2.1 IEMOCAP

This is an audio-visual database that consists of approximately 12 hours of recordings including video, speech, motion capture of the face, and the transliterations corresponding to a total of 10039 recordings [26]. The audios were originally sampled at 16kHz with 16-bit resolution. The database is divided into five recording sessions. Two actors (one male and one female) performed scripted and improvised scenes. The database was annotated by multiple annotators with five emotional labels: anger, happiness, sadness, neutral, and frustration. There are about 10000 samples per class and the annotations are based on the average of the labels assigned by the four labelers.

3.2.2 RAVDESS

This is a multimodal database with emotional speech and songs [49]. Speech recordings of 24 actors (12 male and 12 female) are included. Each actor produced two lexically-matched statements in neutral north American English accent. Seven expressions with different emotional content were produced by the actors: calm, happy, sad, angry, fearful, surprise, and disgust. Besides one expression produced with neutral emotional content, the other expressions were produced twice, with normal and strong level of emotional intensity. There is a total of 1440 recordings sampled at 16kHz with 16-bit resolution. Each class contains 192 samples except the neutral one with only 96 samples.

3.2.3 EMODB

This database contains recordings of 10 German actors (5 male and 5 female) who produced 10 utterances [50]. Seven emotions are labeled in the recordings: anger, boredom, disgust, anxiety, happiness, sadness, and neutral. The recording process was performed in ideal acoustic conditions and using a professional audio setting. The distribution of samples among the emotions is not even, i.e., there are emotions with much less recordings than others.

3.2.4 Konecta voicemails

This database contains voicemails which were recorded at the end of phone-calls between customers and service-agents. In those voicemails the customers give spontaneous evaluations of the service provided by the agent. The customers were informed that their speech was going to be recorded. Due to the nature of the service, it is assumed that the speakers in these recordings are all adults. The audios were recorded at a sampling frequency of 8 kHz and 16-bit resolution. The corpus contains 2364 recordings annotated by experts in QoS (i.e., the labelers listened to each audio file and evaluated whether the customer was satisfied or not). The experts labeled the audios as satisfied, dissatisfied, or neutral based on the linguistic content of the voice-mail given by the customer. That is, when the annotators could conclude that the customer did not receive the expected service, such voice-mail was labeled as dissatisfied. This way of labeling not only takes into account the linguistic content, but also the expert-knowledge about customer service. Only satisfied and dissatisfied categories are considered in this study. There is a significant difference in the length of satisfied and dissatisfied recordings (t-test [51] with $p \ll 0.05$). This is shown in Figure 3.2, where longer recordings for dissatisfied customers can be observed. Although it seems like this information is valuable for some applications, it is important to clarify that, in general, in speech analysis the recordings have to be normalized w.r.t time to avoid biases. For instance, time duration could be very different due to cultural and language differences or could differ among kinds of provided services. Besides, gender-balance is validated through a chi-square test [51] ($p \approx 1$).

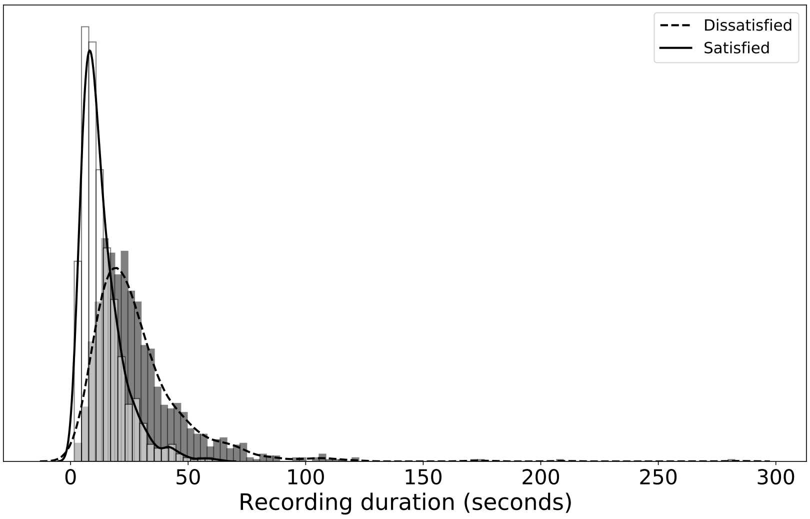


Figure 3.2. Duration distribution for satisfied and dissatisfied classes in the KONECTADB.

Chapter 4

Methods

This chapter describes the methodology followed in this study. The description is divided into three sections: ASR system implementation, emotion classification & CS estimation from speech, and multimodal CS analysis. The first section describes the process to develop an ASR system robust against noise. The second section explains a methodology to find suitable features to robustly recognize emotions and estimate CS from speech in real acoustic scenarios. The third section evaluates different multimodal analysis based on text and acoustic modalities in order to estimate CS.

4.1 ASR system implementation

Figure 4.3 illustrates the overall process to train and test an ASR system. At the top, it is described the training stage, and at the bottom the test one.

4.1.1 Training stage

This stage encompasses feature extraction, Language Model (LM), Acoustic Model (AM), and the Dictionary.

Feature extraction

This study considered a total of 40-MFCCs extracted from 40 triangular Mel-frequency bins with a window size of 25ms and a step size of 10ms. The spectrogram is unit-normalized.

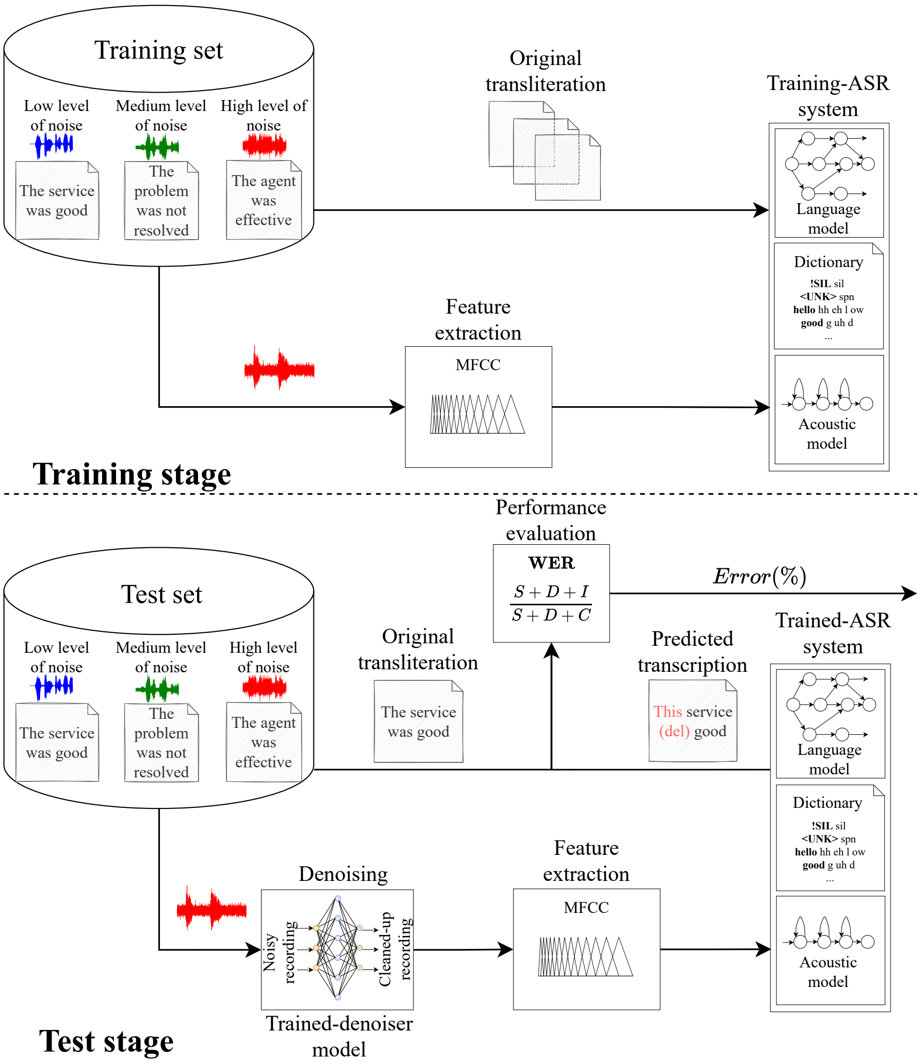


Figure 4.1. General methodology for ASR system implementation.

Language model

The transliteration of the training set was used to train a 3-gram language model. The probabilities of a language model can be computed by counting relative frequencies of the 3−tuples of words that belong to the training set. To estimate the probabilities of the 3−gram model, the following equation is used:

$$P(w_n|w_{n-1}, w_{n-2}) = \frac{C(w_{n-2}, w_{n-1}, w_n)}{C(w_{n-2}, w_{n-1})} \tag{4.1}$$

where w_n represents a word located in position n, and C represents a function that counts the number of occurrences of the word sequence defined in its argument.

Acoustic model

This study considers a 3-state HMM for modeling temporal dependencies. Four different models are trained and evaluated to represent the acoustic distribution of each acoustic unit (HMM state).

- **GMM:** This acoustic model is based on GMM models. A total of 100 thousand of Gaussian components and a decision tree of 4016 leaves were considered in this work. The GMMs were trained using a Maximum Likelihood estimation. This model was also used to force-align the training data and it is regarded as the baseline in this study.
- **TDNN:** This architecture consists of six TDNN layers with 1536 units and a bottleneck dimension of 256. Each layer contains a frame context of three and a skip connection coming from the previous layer's input. The last TDNN layer's output is fed into a fully connected layer with a softmax activation function. Details of this method can be found in [52].
- **LSTM:** This architecture consists of four bidirectional LSTM layers with a *tanh* activation function. Each layer contains 550 units and a dropout regularization of 0.2. The last LSTM layer's output is fed into a fully connected layer with a $softmax$ activation function.
- **GRU:** This architecture consists of five bidirectional GRU layers with a *relu* activation function. Each layer contains 550 units and a dropout regularization of 0.2. The last GRU layer's output is fed into a fully connected layer with a $softmax$ activation function.

The forced-aligned data generated by the GMMs are used to train the DL-based models. On the one hand, the Kaldi toolkit [53] is used to train the TDNN model using Stochastic Gradient Decent (SGD) with an initial learning rate of 0.00015 and batch size of 64. On the other hand, ADAM optimizer with an initial learning rate of 0.0002 and batch size of 64 is used to train the LSTM- and GRU- based architectures using Pytorch-Kaldi framework [54]. Only considered five epochs are considered due to computational constraints.

Dictionary

The dictionary contains the phone pronunciation of each word to be recognized in our model. The phone composition is performed using pronunciation rules of the Spanish language from Colombia. To build the dictionary, the most frequent words seen in the training set were selected. This study considered 20 thousand different words.

4.1.2 Test stage

This involves the same processes as the training stage and also includes denoising and performance evaluation. To avoid any possible bias and to guarantee the generalization capability of the model, this process only considers recordings of the test set.

Denoising model

The denoiser is thought to enhance the speech signals. Thesis used a similar approach as the one presented in [55]. A Short-Time Fourier transform (STFT) was computed using a 25ms Hanning window with a step size of 10ms. The model architecture consists of a fully connected layer followed by two GRU layers and finally, two fully connected layers. The input to the model is the unit-normalized complex spectrogram. The last layer predicts the masking coefficient to denoise the complex spectrogram. The mask aims to reduce noise effects by multiplying weights closer to zero with those frequency bands that contain noise energy. The masked complex spectrogram is then transformed into the time-domain using the inverse STFT function. The complete filter process is illustrated in Figure 4.2.

The augmented training dataset of Konecta calls described in Section 3.1.1 is used to train the system. The original signals are used as the ground truth during the training process of the GRU. The GRU-based denoiser is trained with Pytorch using the Adam optimization strategy with an initial learning rate of 0.0001 and a batch size of 10. Only five epochs are considered due to computational constraints.

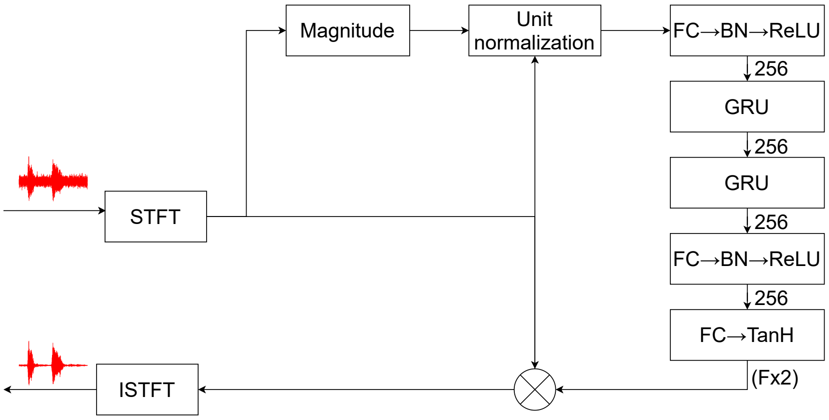


Figure 4.2. Denoising process [55]. F is the number of frequency bins and $\bigotimes$ is the Hadamard product.

Performance evaluation

Once the ASR system is trained, this is used to convert the recordings into text transcriptions of the test set. The Word Error Rate (WER) was computed to evaluate the model. This is the a well known performance measure typically used to evaluate ASR systems [56]. It is defined as follows:

$$WER = \frac{S+D+I}{S+D+C} \tag{4.2}$$

where,

- ✓ S: # of substitutions.
- ✓ D: # of deletions.
- ✓ I: # of insertions.
- ✓ C: # of correctly recognized words.

WER compares two text chains. This metric counts the number of operations needed to convert one text into another one. WER is computed upon the original transcription and the predicted transcription in the case of an ASR system.

4.2 Emotion classification and customer satisfaction estimation from speech

Figure 4.3 illustrates the overall methodology followed for CS estimation, which includes pre-processing, feature extraction, and classification.

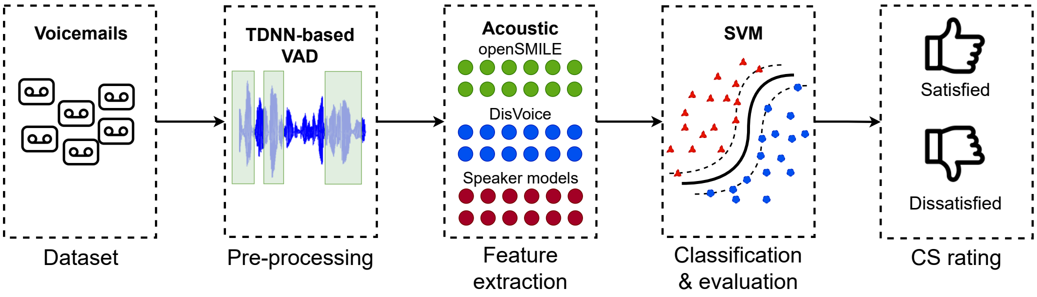


Figure 4.3. General methodology for customer satisfaction estimation from speech.

4.2.1 Preprocessing

VAD is applied to the recordings of the Konecta voicemails to remove long silence segments. The VAD algorithm is a pre-trained model based on time-delay neural networks. The architecture consists of seven time-delay neural layers, a fully-connected layer and an output layer with softmax as the activation function. A total of 20 MFCCs are used as input. To prepare the training and test sets, the Fisher corpus [57] is word-aligned by using a GMM-based ASR system and then the word segments were annotated as speech while the rest segments as silence. Thus, the targets are speech and silence classes. The model[1] was trained using the Kaldi toolkit [58].

4.2.2 Feature extraction

Three different approaches are considered in this study: two speaker models, the openSMILE feature set and the feature sets extracted with DisVoice,

[1] It can be downloaded from: https://kaldi-asr.org/models/m4

which include phonation, articulation, and prosody measures. Each feature set is described below.

i-vectors

Earlier studies on speaker recognition methods to model speaker traits through high-dimensional super-vectors were based on Gaussian Mixture Models (GMMs) adapted from a Universal Background Model (UBM) [59]–[61]. That approach produces a big set of parameters, which requires a large dataset to be trained. To solve this problem, the authors in [62] proposed a low-dimensional vector representation called identity vector (i-vector). The low-dimensional space is defined by a matrix called the total variability matrix T, which models both speaker and channel variability. The new model is represented as follows:

$$M = m + Tw \tag{4.3}$$

where M is the GMM super-vector of a speaker, m is the speaker- and channel-independent super-vector (taken from an UBM super-vector), T is a rectangular matrix of low rank (the total variability matrix) and w is the i-vector, which is a random vector with a standard normal distribution $\mathcal{N}(0, I)$. The concept behind this is that a low-dimensional latent vector (w) exists and represents the characteristics of the speaker. Equation 4.3 can be resolved through joint factor analysis, where w represents the factor of the total variability matrix (T).

This master's thesis considers a total of 1000 recordings with an average duration of 22 seconds per recording to train the UBM and the T matrix. These recording were provided by Konecta Group S.A.S.®. The recordings were randomly selected and do not come from the same speakers of the Konecta voicemails. The implementation of this approach was performed with the Kaldi toolkit [58].

x-vectors

These are Deep Neural Network (DNN) embedding features, which were trained for speaker recognition and verification [63]. Table 4.1 shows details of the architecture of the DNN to extract the x-vectors.

The input features are MFCCs extracted from 24-dimensional filter banks with a frame-length of 25ms and a step-size of 10ms. The spectrogram is

Table 4.1. Embedding DNN architecture [63]. x-vectors are extracted in segment6. N: Number of training speakers.

Layer	**Layer Context**	**Total Context**	**Input x output**
Frame1	$[t-2, t+2]$	5	120x512
Frame2	$\{t-2, t, t+2\}$	9	1536x512
Frame3	$\{t-3, t, t+3\}$	15	1536x512
Frame4	$\{t\}$	15	512x512
Frame5	$\{t\}$	15	512x1500
Stats pooling	$[0, T)$	T	1500Tx3000
Segment6	$\{0\}$	T	3000x512
Segment7	$\{0\}$	T	512x512
Softmax	$\{0\}$	T	512xN

mean-normalized over a sliding window of up to 3 seconds. The first five layers operate at a frame level and build their context according to the previous layer. The stats pooling computes the mean and standard deviation from the output of the Frame5 layer using all T frames of the signal. The mean and standard deviation are concatenated and propagated through the segment-level layers. Finally, the softmax layer predicts the speaker. The output of the segment6 layer is extracted to create the x-vector. The implementation of this approach was also performed with the Kaldi toolkit [58].

The openSMILE feature set

The openSMILE feature set is extracted by following three steps: (1) The 38 low-level descriptors shown in Table 4.2 are computed with a step-size of 10ms and a Hanning window with 25ms of length (except pitch related features, which are extracted from Gaussian windows of 60ms); all instances are smoothed using a moving average filter of 3 frames. (2) Besides the low-level descriptors, 38 first-order regression coefficients are included. (3) A total of 21 statistical functionals shown in Table 4.2 are computed per feature vector. More information and details about how to extract these features can be found in [27].

Table 4.2. The low-level descriptors and the functionals.

Descriptors	**Functionals**
PCM loudness	Position max./min.
MFCC [0-14]	arith. mean, std. deviation
log Mel Freq. Band [0-7]	skewness, kurtosis
LSP Frequency [0-7]	lin. regression coeff. 1/2
F0	lin. regression error Q/A
F0 Envelope	quartile 1/2/3
Voicing Prob.	quartile range 2-1/3-2/3-1
Jitter local	percentile 1/99
Jitter consec. frame pairs	percentile range 99-1
Shimmer local	up-level time 75/90

Phonation, articulation and prosody features

These features are extracted with the DisVoice framework [35], which was originally developed to model neurological disorders and now this study evaluates its suitability to model emotional speech signals and CS. The source code to extract the features presented in this subsection can be found in [35].

Phonation: The phonatory characteristics of a speaker have been typically analyzed in terms of features related to perturbation measures such as jitter (temporal changes in the fundamental frequency), shimmer (amplitude changes in the signal), amplitude perturbation quotient (APQ), and pitch perturbation quotient (PPQ). APQ and PPQ are long-term perturbation measures of the amplitude and the fundamental frequency of the signal, respectively. Fuller et al. found that perturbation measures like jitter are a reliable indicator of stressor-provoked anxiety [64]. Additionally, the results reported in [65] show that jitter and shimmer are useful to model emotion and stress patterns in speech. The phonation features considered in this work include the first and second derivative of the fundamental frequency, jitter, shimmer, APQ, PPQ, and logarithmic energy. These measures are computed upon voiced segments. The global representation per speaker consists of the mean, standard deviation, skewness, and kurtosis of the resulting feature vector.

Articulation: This feature set is inspired in the fact that the transition between voiced and unvoiced segments encodes relevant information about the capability of the speaker to produce well-articulated utterances. Previous studies have shown that this approach is valid for neurological disorders where speech production is affected [35], [66]. Now this study evaluates its suitability to model emotional speech and CS. The main hypothesis is that people under stress due to a bad quality of service are prone to produce more hesitations while speaking. This phenomenon could be associated to abnormal energy patterns in the vicinity of the border between voiced and unvoiced sounds, i.e., around the time when vocal fold vibration starts and/or finishes.

Besides the aforementioned hypothesis, the use of the Teager Energy Operator is introduced. The authors in [67] show that the source of speech production is not actually a laminar airflow. Instead, it consists of vortex-flow interactions with the vocal tract boundaries. There is evidence that shows these air vortices to be generated during the early opening phase and the latter closing phase of the vocal fold, as occurred during the transitions between voiced and unvoiced segments [68]. In addition, it is believed that changes in vocal system physiology induced by stressful and/or fearful conditions such as muscle tension affect the vortex-flow interaction patterns in the vocal tract [69]. These changes directly affect the spectrum of the speech signal. Thus, features sensitive to the presence of these additional vortices between the transitions could help to detect the emotional state of the speaker. In this case the main assumption is that if the speaker's speech is altered due to changes in the emotional state, then such changes will produce abnormal articulation patterns during speech production.

The complete articulation feature set includes the first two formants with their first and second derivatives extracted from the voiced segments. The energy in the transitions between voiced and unvoiced segments which is computed and distributed according to the Bark bands, and MFCCs with their first and second derivatives are also measured in the aforementioned transitions.

Prosody: Two types of prosodic features can be found in the literature: basic and compound. Basic prosody attributes include loudness, pitch, voice quality, duration, speaking rate, and pauses. Variations of these measures over time constitute the compound prosodic attributes, which are intonation, accentuation, prosodic phrases, rhythm, and hesitation [70]. Human beings typically use prosody features to identify emotions present in daily conversations [71]. For instance, in active emotions like anger, pitch and energy are high while they are relatively low in passive emotions like sadness. This paper considers prosodic features based on duration, fundamental frequency, and energy to model emotional patterns in speech. Several statistical functionals are computed per feature vector including mean value, standard deviation, skewness, kurtosis, maximum, and minimum. Details of the methods and algorithms to extract these features can be found in [35].

Although openSMILE and the introduced features with DisVoice share several similarities, there are relevant differences that deserve to be mentioned:

1. Unlike openSMILE, the articulation features are only extracted at the onset/offset transitions, which makes it potentially more generalizable to different languages.

2. The articulation does not include log Mel Frequency Band and LSP Frequency.

3. openSMILE does not include Bark band energy.

4. The prosody features based on duration and energy are not included in openSMILE.

5. Phonation features like APQ, PPQ and the pitch dynamic are not included in openSMILE.

6. Some descriptors of our proposed feature set are related to information about dynamics (first and second derivatives) which are not included in openSMILE.

7. The DisVoice framework has no restrictions for its commercial use while the openSMILE framework does.

In summary, on the one hand, our feature set considers a larger number of descriptors (i.e., 154 features) vs. 38 included in openSMILE. On the other hand, openSMILE considers 21 functionals while our feature set only includes six.

4.2.3 Classification and Evaluation

The classification process is performed with a soft-margin SVM with a Gaussian kernel [72], [73]. This classifier has two parameters, complexity C and the kernel bandwidth γ. These two parameters are optimized in a grid-search up to powers of ten where $C \in [10^{-3}, \dots, 10^{4}]$ and $\gamma \in [10^{-6}, \dots, 10^{3}]$. To avoid biased or optimistic results, a nested cross-validation strategy is followed [74]. Five folds for outer and also for inner cross-validation are considered. Both scenarios, SI and SD, are considered in the experiments about emotion classification. The performance of the two-dimensional classifier is measured in terms of recall on the positive class also known as Sensitivity (SEN), recall on the negative class also known as Specificity (SPE), Weighted Average Recall (ACC), Unweighted Average Recall (UAR), and the Receiver Operating Characteristic (ROC) curve. SEN is the ratio of the correctly predicted positive samples to the total number of positive samples. SPE is the ratio of the correctly predicted negative samples to the total number of negative samples. ACC is the weighted arithmetic mean of the recalls. UAR is the arithmetic mean of the recalls. And ROC is a graph that illustrates trade-offs between true positive rate (SEN) and false positive rate when the decision threshold of the classifier changes its position [75].

4.3 Multimodal analysis of customer satisfaction

Figure 4.4 illustrates the overall methodology followed for multimodal analysis of CS, which includes feature extraction, multimodal fusion, and classification.

4.3.1 Feature extraction

Acoustic and text modalities are used. The acoustic modality considers three different feature sets: x-vector, openSMILE, and articulation. The text modality considers the word2vec representation of the text transcriptions generated by an ASR system. Given that the acoustic representations

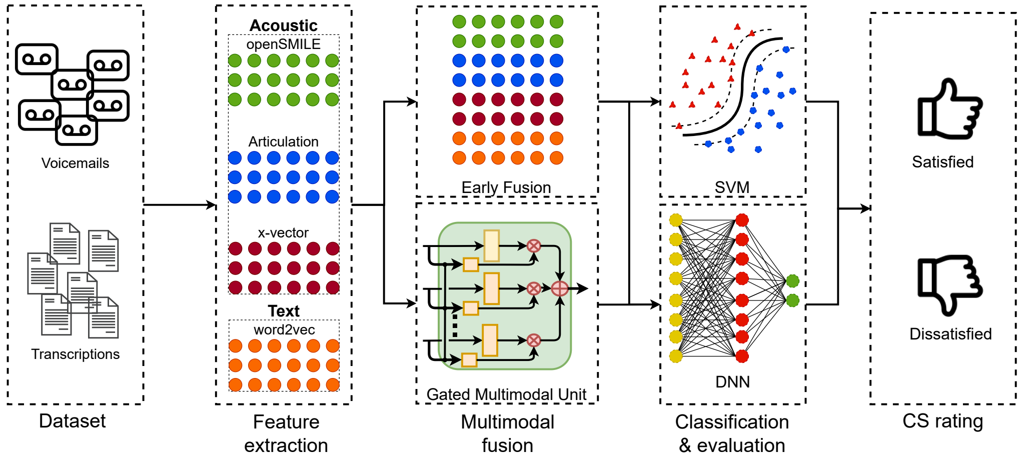


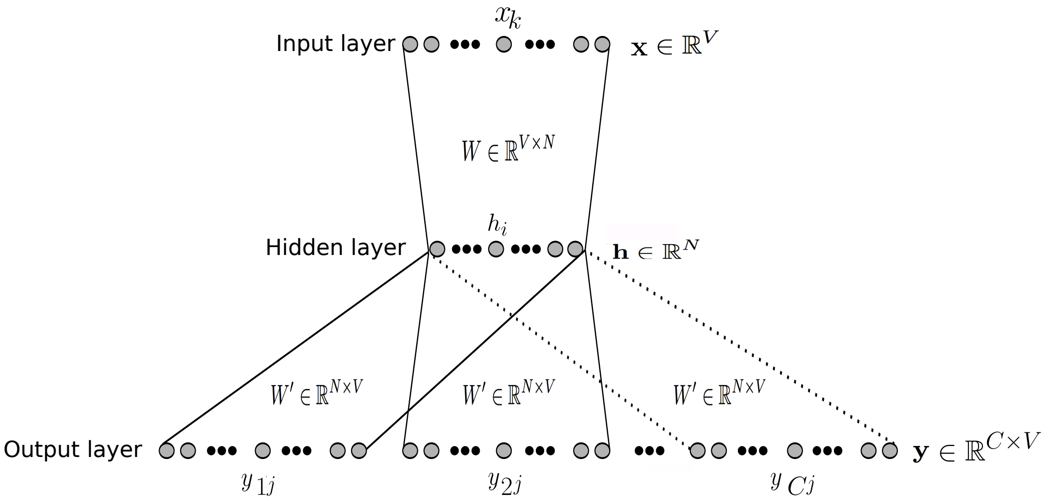
Figure 4.4. General methodology for multimodal analysis of customer satisfaction.

are explained in detail in Section 4.3.1, only text representation is described in this section.

Word2vec

This model takes a large text corpus as input and produces a vector space, typically of several hundred dimensions. Word vectors are positioned in the vector space such that words sharing common context in the corpus are geometrically close to each other [76]. There is a unique vector to represent each word in the corpus. For this reason it is known as a context-independent embedding because the word representation is the same regardless of its context.

This algorithm is based on deep learning to model word relations. The one-hot representations of the words are used as inputs. The word embeddings based on Word2Vec can be calculated by means of two strategies: Continuous Bag Of Words (CBOW) or Skip-Gram. However, this study only considers Skip-Gram. This strategy takes a word as input and the model predicts the context corresponding to such a word. Figure 4.5 shows the neural network structure for the Skip-gram strategy. The input layer consists of the V-dimensional one-hot encoded word vectors. The hidden layer contains N neurons and the output is a V-dimensional vector that corresponds to the target context of the word.



x_k
Input layer
$\mathbf{x} \in \mathbb{R}^V$
$W \in \mathbb{R}^{V \times N}$
h_i
Hidden layer
$\mathbf{h} \in \mathbb{R}^N$
$W' \in \mathbb{R}^{N \times V}$
$W' \in \mathbb{R}^{N \times V}$
$W' \in \mathbb{R}^{N \times V}$
Output layer
$\mathbf{y} \in \mathbb{R}^{C \times V}$
y_{1j}
y_{2j}
y_{Cj}

Figure 4.5. Skip-gram model. Figure adapted from [77].

These type of representations have gained popularity not only because each word-embedding keeps the semantic properties of the word, but also because it is a model trained in an unsupervised manner, i.e., the texts that are analyzed do not require prior labeling. This makes it possible to train models with a large amount of data from freely accessible text resources without spending money on expensive hand-labeled databases.

This work uses a pre-trained Word2Vec with 300 dimensions. The model was trained with the Spanish WikiCorpus, which contains 120 million words [78]. The embedding representation is performed with Skip-Gram strategy and 8 context words.

4.3.2 Multimodal fusion

Two types of fusion schemes are evaluated: EF and GMU. All possible combinations of the feature sets described in Section 4.3.1 are performed.

Early Fusion: In this scheme, features are extracted from each information source yielding an unimodal representation. The extracted features are then combined into a single representation. After combination of unimodal representation, the fused features are fed into a classifier that learns concepts. As depicted in Figure 4.6a, EF integrates the features from the beginning.

Gated Multimodal Unit: This model is intended to be used as an internal unit in a neural network architecture whose purpose is to find an intermediate representations based on the combination of information from different modalities [46]. This model learns how each modality contributes to a particular sample. GMU is based on multiplicative gates and is inspired by the flow control in recurrent neural networks. Figure 4.6b illustrates the structure of a GMU cell for multiple modalities. Each input x_i is a feature vector associated with a modality i. Each vector feeds a hidden layer with a tanh activation, which models an internal pattern based on the particular modality. For each input modality, x_i, there is a gate layer with σ activation, which controls the contribution of each modality to the activation of the GMU, h.

The specific mathematical form of the GMU is thus expressed as followed:

$$h_i = \tanh\left(W_{hi} \cdot x_i\right) \tag{4.4}$$

$$z_i = \sigma(W_{zi} \cdot [x_1, x_2, ..., x_k]) \tag{4.5}$$

$$h = z_1 * h_1 + z_2 * h_2 + ... + z_k * h_k \tag{4.6}$$

where x_i is the feature vector, W_{hi} is the internal encoding matrix, and W_{zi} is the gate matrix for modality i. Additionally, $[\cdot, \cdot]$ represents the concatenation operator.

4.3.3 Classification and Evaluation

This work considers two classifiers, namely an SVM and a fully-connected DNN. The EF approach is evaluated using both classifiers while the GMU approach only considers the DNN. Given that this experiment requires more computation resources due to the feature combination, nested cross-validation is not considered. To reduce the expensiveness of training stage, the data is divided into train (75%), validation (10%), and test (15%) sets. This process is repeated 10 times.

SVM: the same hyperparameters of the SVM, explained in Section 4.2.3, are used. Instead of using nested cross-validation, the data are split in training (75%), validation (10%) and test (15%).

DNN classifier: In general, the architecture of the DNN classifier consists of a fusion (input) layer, a hidden layer (256 units), and an output layer with a softmax activation function for classification. In the case of gated

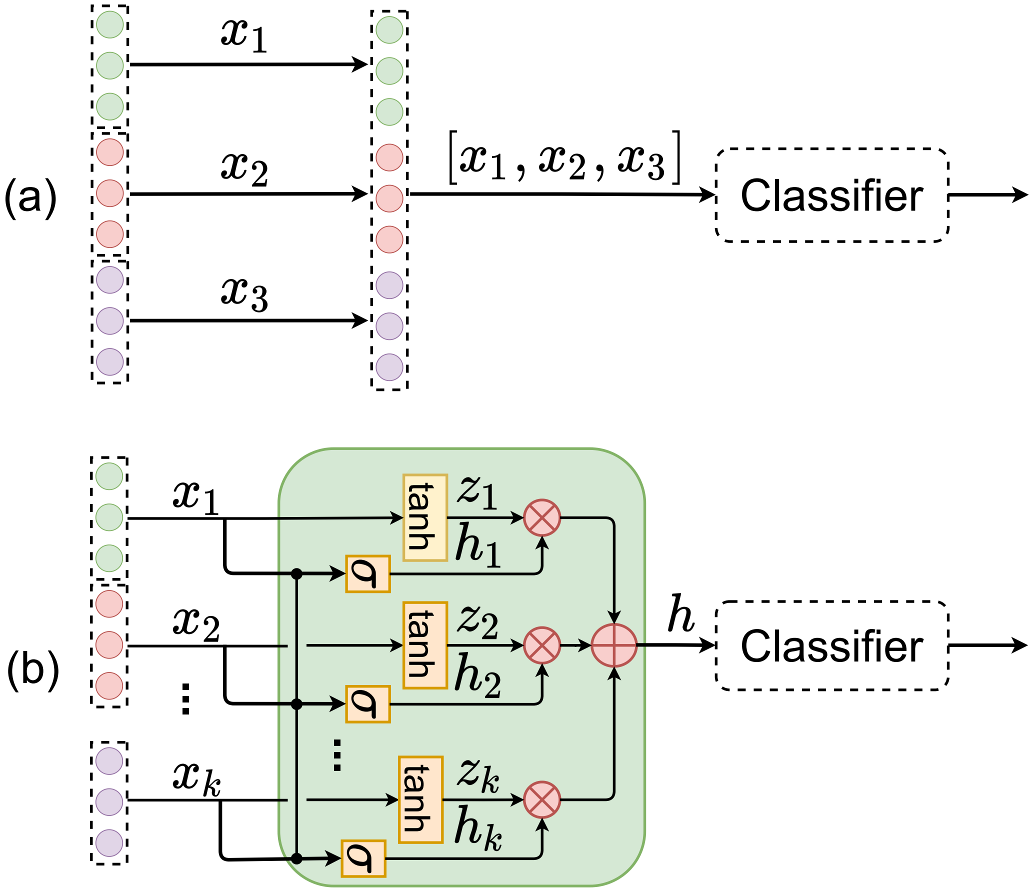


Figure 4.6. Illustration of multimodal fusion approaches. a) Early Fusion approach using three modalities. b) Gated Multimodal Unit [46] approach using k modalities. $[\cdot, \cdot]$ represents the concatenation operator

units, the fusion layer is the output of the GMU model. A ReLU activation function is used in both the fusion layer (for the EF approach) and hidden layer. A batch normalization and dropout are applied after the fusion layer.

The DNN is trained using PyTorch [54] with 10 epochs, an initial learning rate of 0.0001, and a batch size of 64. For optimization, Adam optimizer without weight decay is used. As a loss function, a binary cross-entropy with logits is applied for numerical stability. This model is evaluated using the same data split of the SVM. The process is also repeated 10 times.

Chapter 5

Experiments and Results

This chapter presents the results obtained for the three steps: results of ASR, results of speech emotion classification and & CS estimation, and results of multimodal analysis of CS.

5.1 Results of ASR systems

With the aim to develop a robust ASR system, four different acoustic model architectures are trained and evaluated in non-controlled acoustic scena-rios. The following are the models: (1) GMM-based model, (2) TDNN-based model, (3) LSTM-based model, and (4) GRU-based model. Finally, a DL-based denoiser is implemented to improve the recognition performance.

5.1.1 Results of acoustic model

The call center database described in Section 3.1.1 is used to train each ASR system. The speed-perturbed training sets (LN, MN, and HM) described in Section 3.1.4 are mixed during the training. The models are evaluated in each real acoustic scenario. Table 5.1 shows the performance of the different ASR systems for each scenario. Note that all DL-based models outperform the baseline (based on GMMs). The LSTM model yields the best performance in non-controlled acoustic conditions with WER values of $22,55\%$ and $27,99\%$ for MN and HN scenarios, respectively. Note that all models except the GMM-based one, obtain similar WER values in the LN condition, that is: $21,73\%$ for TDNN, $21,31\%$ for LSTM, and $21,30$ for GRU.

Table 5.1. Performance of the ASR systems in terms of the WER in each real acoustic conditions. **LN:** Low level of noise. **MN:** Medium level of noise. **HN:** High level of noise.

Architecture	Acoustic scenario		
	LN	MN	HN
GMM	32,10	35,54	52,47
TDNN	21,73	23,48	30,94
LSTM	**21,31**	**22,55**	**27,99**
GRU	**21,30**	22,67	28,77

5.1.2 Results of denoising process

The denoiser described in Section 4.1.1 is trained to enhance noisy speech signals. The training set augmented with additive noise is used to train the model. Two test sets are considered to evaluate the capability of the filter to suppress/remove the noise: (1) The artificially created noisy recordings (the scenario described in Section 3.1.3), and (2) The noisy recordings of the HN test set (real scenario). WER values of the ASR systems are computed for the noisy and enhanced speech signals for comparison purposes. Table 5.2 shows the performance obtained for the DL-based ASR systems in the simulated and real scenarios. The TDNN model shows improvements in both scenarios when the denoiser is applied. In the simulated conditions, the WER goes down from 40,41% to 35,70%, and in the real noisy conditions it changes from 30,94% to 26,83% which is actually the best performance obtained for noisy conditions. For the case of the LSTM-based model in the simulated scenario, without denoising it yields the worst WER for noisy conditions (44,39%), but it improves to 38,88% after applying the denoiser. Although the improvement is relatively high (5,51 absolute percentage points), the result is still the worst among the rest obtained in that scenario. Regarding its results in the real conditions, without any denoising procedure, the LSTM yields the best WER (27,99%), however, when the denoiser is applied the WER value increases to 29,63%. A similar behavior can be observed for the GRU model, where the WER value obtained in the simulated conditions prior to the denoiser is 40,41% and it gets better to 37,27% when the denoiser is applied; however, in the real noisy conditions, its WER value gets worst in 1 absolute percentage when the denoiser is applied (from 28,77% to 29,77%).

Table 5.2. Performance of the ASR systems in terms of the WER before and after applying the denoiser. **Simulated:** The augmented test set. **Real:** The HN test set of Konecta calls. Values in %.

Model	Simulated conditions		Real conditions	
	Noisy	Enhanced	Noisy	Enhanced
TDNN	**40,41**	**35,70**	30,94	**26,83**
LSTM	44,39	38,88	**27,99**	29,63
GRU	40,41	37,27	28,77	29,77

5.1.3 Discussion

Regarding the results obtained by the LSTM in the acoustic models' evaluation, this architecture seem to be the best in different acoustic environments. However, once the denosier is implemented and the noisy recordings pass through it, the LSTM performance decreases. Additionally, this model is the most affected after the recordings are contaminated with out-of-domain additive noise. Its WER is 44.49%, being 4% above the TDNN. GRU has similar behavior but to a lesser degree. The case is different for TDNN, having a significant improvement after applying the denoising process. This leads us to conclude that the initial results of the recurrence-based models respond to a bias towards the acoustic training conditions, which explains why they are slightly better than the TDNN in the high noise scenario. Since TDNN is the least affected by the out-of-domain additive noise and it achieves significant improvement with the denoiser, it is possible to say that this model is more robust under different acoustic environments than the other architectures. For this reason, this model is chosen to generate the text transcriptions for the text modality in the multimodal CS analysis experiment.

5.2 Results of speech emotion classification and customer satisfaction estimation

Different experiments are performed with the different datasets considered in this study. Experiments with IEMOCAP, EMODB, and RAVDESS are all multi-class, i.e., all emotions included in these corpora are considered. The experiments with the Konecta voicemails corpus are bi-class, where the aim is to discriminate between satisfied vs. dissatisfied customers. The results are reported in terms of UAR and ACC. Optimal hyper-parameters found for each classification experiment are also reported to allow direct comparisons in future studies. Experiments using individual features and also their combinations are included. The symbol + in the feature representation means fusion with other feature sets, while art, pro, and pho are articulation, prosody, and phonation, respectively. Due to the fair performance of the i-vectors, they are not included in the fusion experiments. Besides, a reduced version of the openSMILE is considered by applying Principal Component Analysis (PCA) over the original feature set. This experiment is denoted as openSMILE$_{\mathrm{PCA}}$.

5.2.1 Experiments with IEMOCAP

Table 5.3 shows the performance of the classifier on the IEMOCAP database. The best results considering individual sets of features are obtained with openSMILE and openSMILE$_{\mathrm{PCA}}$ (see Table 5.3). The second best approach is based on x-vectors, with UARs of 57.6% and 65.4%, for SI and SD, respectively. The third best model is the articulation which achieved 54.0% and 59.5% of UAR in the SI and SD scenarios, respectively.

On the other hand, when openSMILE and x-vector are combined, the performance improves 1.0% absolute UAR w.r.t the result obtained with openSMILE$_{\mathrm{PCA}}$ for SI. Nevertheless, the number of features increases from 297 to 2094.

5.2.2 Experiments with EMODB

The results obtained with the EMODB corpus are shown in Table 5.4. Among the individual feature sets, openSMILE yields the best result in the SI scenario, while x-vectors are the best for SD experiments, which is expected considering the well-known capability of them to model specific

Table 5.3. Results of multi-class classification of emotions of the IEMOCAP database. **ACC**: Accuracy. **UAR**: Unweighted Average Recall. The performance metrics are given in [%]

Feature set	**# Features**	**Speaker independent**				**Speaker dependent**			
		C	γ	**UAR**	**ACC**	C	γ	**UAR**	**ACC**
openSMILE	1582	10	0.001	58.9	57.4	1	0.001	68.1	67.2
openSMILE$_{\textbf{PCA}}$	297	1	0.001	**59.8**	**57.4**	1	0.001	**68.6**	**67.0**
i-vector	128	1	0.01	53.5	51.1	1	0.01	60.2	57.5
x-vector	512	1	0.001	57.6	55.9	1	0.001	65.4	63.9
articulation	488	1	0.001	54.0	51.9	1	0.001	59.5	57.2
prosody	103	1	0.0001	44.5	41.2	100	0.0001	48.1	45.2
phonation	28	1	0.001	46.0	43.0	100	0.001	48.5	45.4
art+pro	591	1	0.001	56.1	54.2	1	0.001	60.7	58.5
art+pho	516	1	0.001	56.0	53.6	1	0.001	59.9	57.6
pro+pho	131	1	1e-5	47.5	44.9	10	0.0001	50.4	47.6
art+pro-pho	619	1	0.001	57.0	54.7	1	0.001	61.3	59.3
x-vector+art+pro+pho	1131	1	0.001	59.9	58.4	1	0.001	66.6	65.5
openSMILE+art+pro+pho	2201	10	0.001	58.4	57.7	10	0.001	66.5	66.0
openSMILE+x-vector	2094	10	0.0001	**60.8**	**58.9**	1	0.001	**68.5**	**67.5**
openSMILE+x-vector+art+pro-pho	2713	1	0.0001	60.6	58.3	10	0.0001	68.0	67.1

Note that since there are four classes, the chance level is 25%

characteristics of a given speaker [79]. When features are combined, the best models are those that include openSMILE. The UAR values obtained with openSMILE+x-vector+art+pro+pho are about 5.3% above those obtained with openSMILE individually, in both SI and SD scenarios. It can also be observed that combining either x-vector or art+pro+pho with openSMILE, it helps in improving the classification performance.

Some studies reviewed in Section 1.2.2 reported results on EMODB, but such results were optimistic because they considered the same set to train and test the classifier. For example, the authors in [21], [23] used an SVM to do the classification and reported accuracies above 79% on the SI scenario (similar to our 79.9% ACC). However, these results were achieved on the same set that was used to optimize the classifier. Instead, this study followed a nested cross-validation strategy, where the test set is unseen by the optimal parameters, leading to more realistic and unbiased results.

Table 5.4. Results of multi-class classification of emotions of the EMODB database. **ACC**: Accuracy. **UAR**: Unweighted Average Recall. The performance metrics are given in [%]

Feature set	**# Features**	**Speaker independent**				**Speaker dependent**			
		C	γ	**UAR**	**ACC**	C	γ	**UAR**	**ACC**
openSMILE	1582	10	0.0001	**74.0**	**73.3**	10	0.001	84.6	85.6
openSMILE$_{\text{PCA}}$	157	1	1e-5	57.2	56.0	10	0.0001	82.7	83.1
i-vector	128	10	0.01	59.6	60.7	10	0.01	73.6	74.2
x-vector	512	10	0.0001	60.9	61.4	10	0.0001	**86.0**	**86.0**
articulation	488	1	0.001	58.9	62.7	10	0.001	69.3	71.2
prosody	103	10	0.01	56.4	58.8	10	0.01	54.7	56.9
phonation	28	1000	0.001	50.3	52.2	10	0.01	52.5	53.9
art+pro	591	10	0.001	64.0	67.6	1	0.001	59.3	61.2
art+pho	516	10	0.0001	61.5	65.0	1	0.001	60.7	63.5
pro+pho	131	1	0.001	53.9	55.1	1	0.01	58.9	62.7
art+pro+pho	619	10	0.0001	55.1	57.7	1	0.001	50.5	50.7
x-vector+art+pro+pho	1131	10	1e-4	70.5	73.0	10	1e-4	84.2	84.5
openSMILE+art+pro+pho	2201	10	1e-4	73.3	75.1	10	1e.4	87.2	88.4
openSMILE+x-vector	2094	10	1e-4	78.2	78.8	10	1e-5	87.0	87.5
openSMILE+x-vector+art+pro+pho	2713	10	1e-4	**79.9**	**80.7**	10	1e-4	**90.7**	**91.4**

Note that since there are seven classes, the chance level is 14.3%

5.2.3 Experiments with RAVDESS

Table 5.5 shows the results obtained with the RAVDESS database. The best models of individual features in the SI scenario are obtained with openSMILE, x-vectors, and openSMILE$_{\text{PCA}}$, with UARs of 58.7%, 58.6%, and 57.8%, respectively. In the SD scenario the best model is the one with on x-vectors with an UAR of 83.4%. Regarding the experiments with combined features, the highest UAR is achieved with openSMILE+x-vector+art+pro+pho, which improved from 58.7% (openSMILE) to 63.7% in the SI scenario. In the SD scenario this combination also yields the highest

UAR.

There are several works, most of them based on deep learning, that achieve accuracies of up to 70.0% in the SI scenario using the RAVDESS database [30], [31], [80]. However, it is not possible to know whether those results are based on unseen data. In the SD scenario, x-vector achieved better performance (83.4% ACC) than those obtained by the authors in [31] (82.41% ACC), where a more complex model was proposed.

Table 5.5. Results of multi-class classification of emotions of the RAVDESS database. **ACC**: Accuracy. **UAR**: Unweighted Average Recall. The performance metrics are given in [%]

Feature set	**# Features**	**Speaker independent**				**Speaker dependent**			
		C	γ	**UAR**	**ACC**	C	γ	**UAR**	**ACC**
openSMILE	1582	10	1e-5	**58.7**	**57.6**	10	1e-4	70.9	70.5
openSMILE$_{\mathrm{PCA}}$	240	10	1e-5	57.8	57.2	10	1e-4	77.7	77.6
i-vector	128	1	0.01	44.7	45.3	10	0.01	64.6	64.1
x-vector	512	10	0.001	58.6	58.7	10	0.001	**83.4**	**83.5**
articulation	488	10	0.001	43.6	44.4	1	0.001	46.4	46.6
prosody	103	1	0.01	37.0	36.5	10	0.01	39.1	38.8
phonation	28	1e5	1e-4	34.1	33.5	100	0.001	37.8	36.8
art+pro	591	10	0.001	46.9	48.1	10	0.001	53.8	54.5
art+pho	516	1	0.001	46.9	47.2	10	0.001	46.9	47.4
pro+pho	131	10	0.01	39.0	39.0	10	0.001	40.0	39.9
art+pro+pho	619	10	0.001	47.1	47.8	10	0.001	53.9	54.4
x-vector+art+pro+pho	1131	10	0.001	54.9	55.6	10	0.0001	73.1	73.9
openSMILE+art+pro+pho	2201	10	1e-4	60.7	61.2	10	1e-4	76.2	76.2
openSMILE+x-vector	2094	10	1e-4	61.9	62.7	10	1e-4	82.6	83.1
openSMILE+x-vector+art+pro+pho	2713	10	1e-4	**63.7**	**63.8**	10	1e-4	**82.9**	**82.8**

Note that since there are eight classes, the chance level is 12.5%

5.2.4 Experiments with Konecta voicemails

All recordings in the Konecta voicemails corpus were collected from different customers and only the SI scenario is addressed. Table 5.6 shows the results obtained when classifying between satisfied and dissatisfied customers.

The experiments with individual feature sets show that articulation achieves the best performance, with 74.2% UAR, and the second best UAR is obtained with the openSMILE$_{\text{PCA}}$ model. Note that the articulation feature set has the additional advantage of providing balanced results in terms of SPE and SEN. For the openSMILE model, the difference between SEN and SPE is 20.1%, while for articulation the difference is only 3.6%. This likely indicates that there is a bias in the openSMILE model towards the detection of dissatisfied customers.

When considering the fusion of features, openSMILE+x-vector yields the best result with an UAR of 75.3%. This is because in the new representation space both feature sets complement the CS analysis. Note that this performance is similar to the one obtained with only the articulation features. Similarly to what is observed in the experiments with individual features, the difference between SEN and SPE is higher with openSMILE than with articulation.

Besides numerical results, ROC curves are included in Figure 5.1 with the aim to show the results more compactly. Only the best two models are considered for comparison purposes. Note that the overall performance is similar in both cases.

Discussion

According to the results obtained in the standard databases, it can be noted that the introduced feature sets do not yield satisfactory results when more than four categories are considered. Although the articulation features perform similarly to openSMILE in IEMOCAP, they are up to 10% of accuracy below in RAVDESS and EMODB. Regarding the description of the databases, both RAVDESS and EMODB contain more speakers and emotional categories than IEMOCAP. It is well known that emotions share acoustic patterns that make their differentiation difficult. We believe that our feature set performs poorly considering these two databases due to the variability of speakers and emotions. That is, it is not discriminative enough to deal with such variabilities. In addition, by combining our feature set with other representations, improvements in classification performance are achieved. Although our feature set perform poorly when used independently, its combination with other features complements emotion modeling by generating a new representation space.

Table 5.6. Results of the CS classification (satisfied vs. dissatisfied) in the Konecta voicemails database. **ACC**: Accuracy. **UAR**: Unweighted Average Recall. **SEN**: Sensitivity. **SPE**: specificity. The performance metrics are given in [%]

Feature set	**# Features**	C	γ	**UAR**	**ACC**	**SEN**	**SPE**
openSMILE	1582	10	1e-4	72.8	73.6	82.9	62.8
openSMILE$_{\text{PCA}}$	218	1	1e-4	73.1	73.8	81.9	66.4
i-vector	128	100	0.001	59.6	61.2	62.8	60.8
x-vector	512	10	1e-4	67.3	67.6	72.5	62.0
prosody	488	1	1e-4	66.0	66.4	71.2	60.7
articulation	103	0.1	0.001	**74.2**	**74.3**	**76.0**	**72.4**
phonation	28	100	0.001	67.0	67.4	73.4	60.6
art+pro	591	0.1	0.001	68.9	69.7	80.1	57.7
art+pho	516	0.1	0.001	68.8	69.0	71.4	66.2
pro+pho	131	2000	1e-5	64.1	63.2	51.3	77.0
art+pro+pho	619	1	0.01	73.6	73.8	77.3	69.9
x-vector+art+pro+pho	1131	1	1e-6	69.3	68.8	61.9	76.7
openSMILE+art+pro+pho	2201	10	1e-4	74.6	74.8	77.4	71.7
openSMILE+x-vector	2094	10	1e-6	**75.3**	**75.7**	**80.8**	**69.9**
openSMILE+x-vector+art+pro+pho	2713	100	1e-5	74.6	74.9	78.7	70.6

The aforementioned results obtained with the Konecta voicemails supports the fact that articulation is the best alternative to evaluate CS in real-world environments. This is because customers under stress are more prone to produce hesitations and muscle tension within the vocal tract, which in turn produces abnormal articulation patterns that are reflected in the spectrum of the speech signal. Furthermore, this method is the most appropriate in industrial applications not only for its performance but also because it can be used with any licensing for its commercial use.

The results indicate that the introduced approach is more robust against non-controlled acoustic conditions, which are realistic and may include channel distortions, microphone imperfections, and highly variable acoustic conditions.

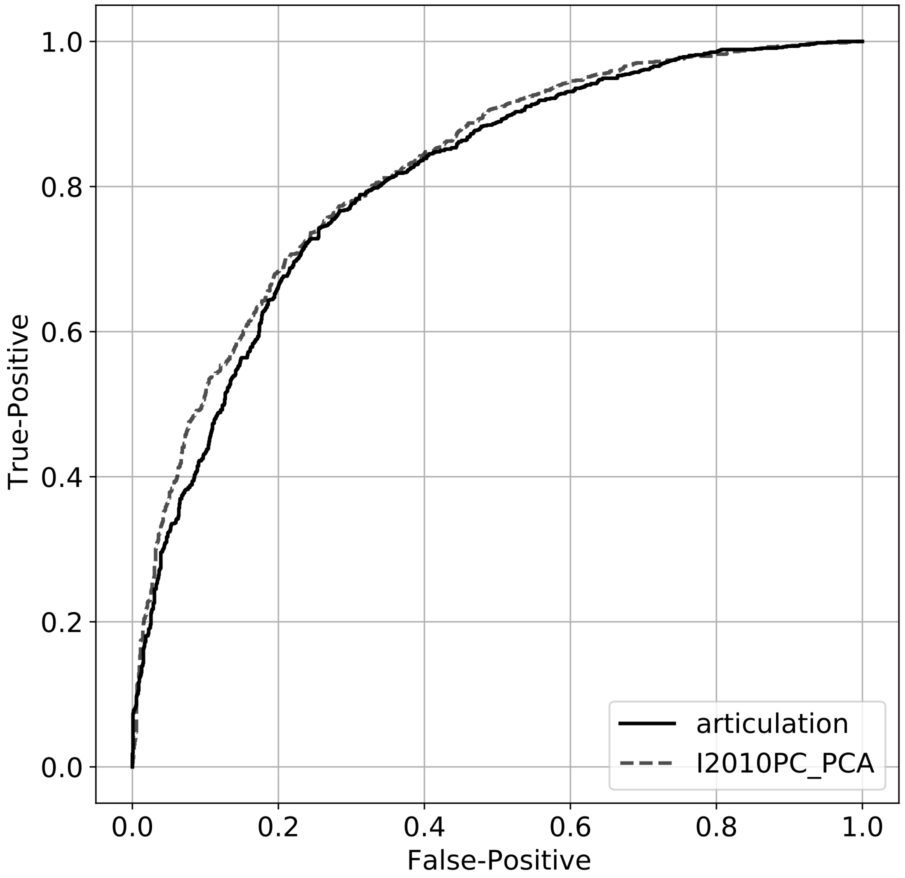


Figure 5.1. ROC curves obtained with articulation feature sets in the classification of satisfied vs. dissatisfied customers.

5.3 Results of multimodal analysis of customer satisfaction

The Konecta voicemails corpus is split in three subsets: train, validation, and test. The splits are stratified so that train, validation and test subsets contain 75%, 15%, and 10% samples of each class respectively. Based on the feature sets described in Section 4.3.1, the multimodal analysis is performed by combining the features using two strategies: EF and GMU. Unimodal analysis is also considered as the baseline. To rate CS, SVM and DNN classifiers are used. Since gated units are based on neural networks, GMU approach is only considered in the DNN classifier. The setup described in Section 4.3.3 is performed to train both classifiers. In the training process, the training subset is used to optimize the model parameters while the validation subset is used to find hyper-parameters and select the best model in terms of accuracy. The optimized model is then evaluated on the test subset. Since each sample belongs to a different speaker, the experiments are speaker independent. The text transcriptions are generated by the best ASR system (i.e., HMM-LSTM). In this section, w2v means word2vec representation.

5.3.1 Results of Unimodal Analysis

Table 5.7 shows the performance of the SVM and DNN classifiers for each unimodal representation. As shown, the best performance is 88.5% accuracy, which is obtained by using the DNN classifier with word2vec, followed by SVM (87.9%) using the same feature set. When considering acoustic features (articulation and openSMILE), the performance is comparable for both classifiers. Similar to the results obtained previously, using articulation features as input resulted in more balanced detections in terms of SEN and SPE. The x-vector feature set reflects the worst performance for both classifiers. In general terms, the DNN-based classifier is slightly better than SVM and requires less computational cost for training due to its simplicity.

5.3.2 Results of Multimodal Analysis

For this case, the EF strategy is applied along with the SVM and DNN, while the GMU technique is only considered in the DNN. The results obtained with the multimodal analysis are shown in Table 5.8. According to the results, all possible combinations of acoustic features (articulation, openSMILE, and x-vectors) improve classification performance over unimodal classification. For example, open+x-vector yields a performance of 76.2%, 75.5%, and 75.9% for EF-SVM, EF-DNN, and GMU, respectively; while openSMILE achieves 74.1% with DNN. As expected, the best results are achieved considering combinations with word2vec as it is the most representative feature according to the unimodal results. The best performance is achieved by w2v+x-vector in both EF-DNN and GMU (88.2% and 88.0%, respectively). However, the

Table 5.7. Results of the CS classification (satisfied vs. dissatisfied) in the Konecta voicemails database for unimodal analysis. **ACC**: Accuracy. **SEN**: Sensitivity. **SPE**: specificity. The results are given in [%]

Feature set	SVM			DNN		
	ACC	SEN	SPE	ACC	SEN	SPE
articulation	73.5	75.9	70.8	74.0	75.4	72.5
x-vector	66.3	70.6	61.3	67.0	73.7	59.3
openSMILE	74.2	79.5	68.1	74.1	77.3	70.5
word2vec	87.9	88.9	86.8	**88.5**	**89.8**	**87.0**

multimodal results do not yield an improvement over the unimodal case. (88.5% for w2v).

What makes it the multimodal analysis powerful is that the modes contain complementary information, so that if one of the modes fails or does not contain pattern information, it is possible to perform the classification with the other remaining modes. Conversely, if the only representation in unimodal analysis is absent, the only possible alternative is the classification by chance. These absences of modalities may occur in real-world applications. For example, if CS is analyzed in chat-based customer opinions, the classification would only be subject to textual analysis (i.e., absence of the acoustic modality). Or for example, if the organization, for some reason, has trouble performing text transcription of the voicemails, the CS classification would only be subject to acoustic analysis (i.e., absence of the textual modality). To demonstrate the multimodal analysis powerful, the absence of one of the modes is simulated in the bimodal analysis. The experiment consists of turning off one of the modalities (i.e, converting it to a vector of zeros) and performing the classification with the remaining representation. In this way, the aforementioned scenarios are being simulated. Only DL-based models are considered since they obtained better results than SVM. It is important to clarify that the models are not retrained, they are only reevaluated with the absence of one of the modes using the test set. Since the models are originally trained with both modalities, they require both modalities for clas-

Table 5.8. Results of the CS classification (satisfied vs. dissatisfied) in the Konecta voicemails database for multimodal analysis. **ACC**: Accuracy. **SEN**: Sensitivity. **SPE**: specificity. The results are given in [%]

Feature set	EF-SVM			EF-DNN			GMU		
	ACC	SEN	SPE	ACC	SEN	SPE	ACC	SEN	SPE
art+open	75.0	79.4	69.9	75.6	79.5	71.0	74.9	78.6	70.6
art+x-vector	75.5	76.0	74.9	75.2	74.0	76.7	75.3	76.2	74.3
open+x-vector	76.2	80.2	71.6	75.5	78.6	71.9	75.9	77.9	73.6
w2v+art	87.8	88.9	86.6	87.5	89.5	85.1	87.5	89.5	85.1
w2v+x-vector	87.5	89.2	85.5	**88.2**	**90.5**	**85.5**	88.0	89.5	86.2
w2v+open	87.2	89.4	84.8	87.5	89.7	85.0	87.5	89.7	85.0
art+open+x-vector	76.4	79.5	72.8	76.5	78.3	74.5	76.8	78.6	74.7
art+open+w2v	86.6	89.4	83.3	85.7	87.9	83.1	86.4	86.7	86.1
open+x-vector+w2v	87.0	89.4	84.2	87.1	88.9	85.1	87.9	88.6	87.2
x-vector+open+art+w2v	86.9	89.4	84.0	86.9	88.9	84.6	86.5	87.5	85.3

sification. For this reason, to represent the absent modality, a vector having the same dimensionality as the original vector is required. This study represents the absent modality with a vector of zeros, which is entered into the model along with the other modality. Figure 5.2 shows the ROC curves for open+w2v combination. Other bi-modal fusions were not included because they did not yield satisfactory results in this experiment. *off* means that the modality is absent (i.e, it is a vector of zeros). So, GMU(open_off+w2v) means that the classification is performed through the GMU-based model using the feature fusion between openSMILE and word2vec, with openSMILE being absent. Note that when one of the modalities is absent, the model can still classify better than the change level. For instance, the absence of the w2v representation produced a 62.7% and 60.2% accuracy for GMU and EF-DNN respectively. As expected, the performance of the model is good even when openSMILE is absent, because the remaining features (w2v) is the most representative according to previous experiments. These experiments demonstrate that the modes complement each other in order to perform more robust classification.

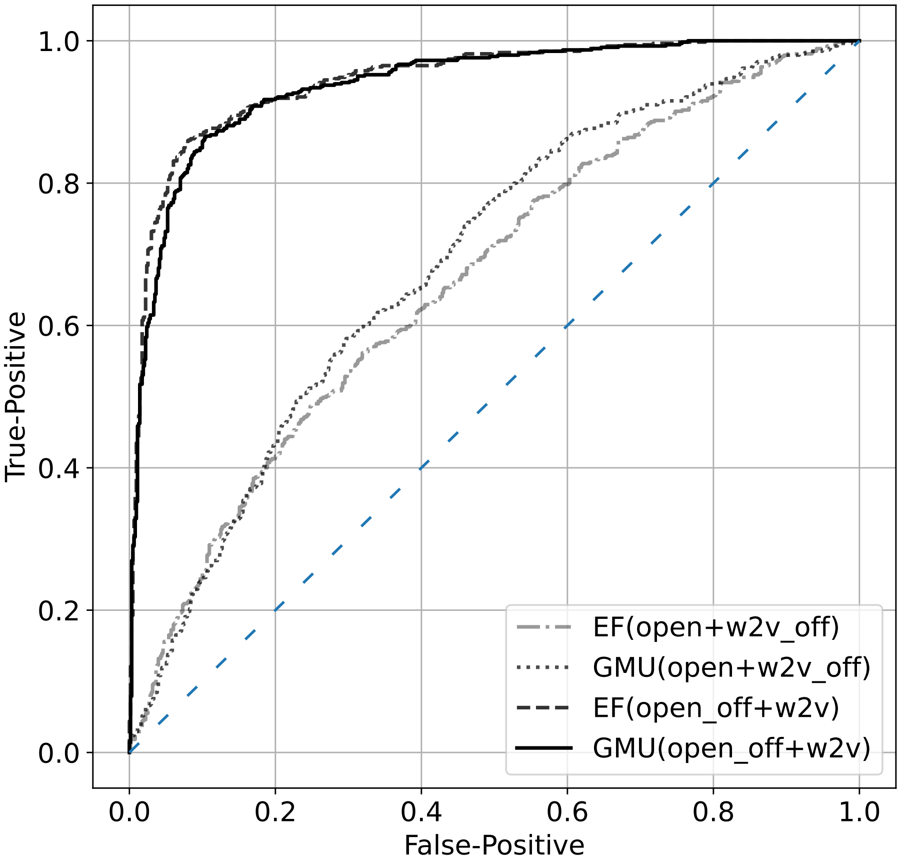


Figure 5.2. ROC curves obtained with open+w2v combination by turning off one of the modes. *off* means that the mode is absent.

Discussion

The good performance obtained in the unimodal analysis by the text representation is due to the keywords found in each category. From experience, we know that satisfied customers usually pronounce positive words such as “thanks”, “nice” or “good”. In addition, their message tends to be short and limited in vocabulary. On the contrary, a dissatisfied customer tends to give details about how bad the service was. The vocabulary is broader and related to service or business concepts. In addition, there are words that reflect dissatisfaction such as “problem”, “inconvenient” or “bad”. These particular characteristics of each class make classification by text more accurate than by acoustic.

Regarding the multimodal analysis, all the combinations that consider word2vec representation achieve the best performance as expected. No combination generates better results than those obtain by unimodal analysis. Although multimodal analysis is more complex, it is also more robust due to the complementary information from each sources. If one of the modes fails, the model can continue the analysis with the remaining modes. This capability can be seen in the ROC curve shown in Figure 5.2, where the feature sets are turned off before entering the classifier. All experiments produce better results than chance classification. This indicates that each mode is contributing to representing the emotional pattern. In addition, it is important to mention that GMU achieves better performance than EF-DNN in this experiment, indicating that this architecture better prioritizes one mode or the other to model the pattern properly.

Chapter 6

Conclusions and future work

This chapter contains the conclusions and future work potentially derived from each experiment.

6.1 Conclusions about ASR system

This experiement presented a methodology to improve the recognition performance of ASR systems. Four different acoustic models are trained and evaluated in non-controlled acoustic conditions: (1) GMM-based model (Baseline), (2) TDNN-based model, (3) LSTM-based model, and (4) GRU-based model. The models were trained with recordings of a call center database, called Konecta calls. This database contains customer service telephone calls. Each recording was captured in real acoustic conditions and it was labeled in terms of its level of noise: low, medium and high. These acoustic conditions allowed to evaluate the models in real noisy acoustic conditions. The hybrid-LSTM model achieved the best performance for medium and high levels of noise. However, this was true when acoustic evaluation conditions were the same as the training ones.

With the aim to improve the recognition performance, a DL-based filter was developed to clean the speech signals. The portion of Konecta calls with low level of noise was artificially contaminated with noise signals taken from Demand Noise Dataset. The denoiser was trained using the noisy recordings. Once the denoiser was trained, the ASR models were again evaluated in two scenarios: (1) Simulated (The artificially contaminated test set), and (2) the real test set with the recordings originally labeled as high level of noise. The WER was computed before and after passing the recordings through the denoiser. In real conditions, the LSTM and GRU did not improve their performance after passing the noisy recordings through the denoiser. In the simulated conditions, both models were the most affected when the clean recordings were contaminated with out-of-domain additive noise. Therefore, these recurrence-based models were over-fitted to the original channel conditions since they obtained the worst performance after such conditions changed. This is why they were better than TDNN in the first experiment. On the other hand, the TDNN model achieved the best results when the denoiser was applied in both simulated and real acoustic scenarios. There was an absolute improvement of 4.11% in the real scenario. This result indicated that the denoiser could enhance the speech signal even in noisy conditions never seen during training. In addition, TDNN was more robust to channel changes imposed by the out-of-domain additive noise or the natural denoiser perturbations, indicating that this model generalizes better to different acoustic conditions. For future work, more complex architectures will be explored in the denoising process to see whether the performance can be further improved.

6.2 Conclusions about CS evaluation

This experiment evaluates different "standard" feature sets typically used to classify emotions in speech, and also presents a novel approach based on modeling phonation, articulation and prosody aspects that may vary when the emotional state of the speaker is changed. The methods are also evaluated in the problem of classifying customer satisfaction based on speech recordings where customers give their opinion about the received service. The databases with emotional speech recordings are those typically used in the literature, which consider acted emotions, limited number of speakers, and relatively controlled acoustic conditions. Conversely, the database recorded in the call-center was collected without any control over the communication channel or microphone, and considers recordings of real opinions (i.e., non-acted) of the customers about the received service. The results show that openSMILE features achieved the best results when classifying emotional speech. Regarding the speaker models, x-vectors outperformed the rest of approaches in two of the three databases where the SD scenario was considered. This was expected since this method has shown excellent results in modeling speaker-specific information.

The features included in our proposed approach were in the top-three of the best feature sets in IEMOCAP, but did not yield satisfactory results in the EMODB and RAVDESS datasets in which openSMILE was 10.6% and 13.2% accuracy higher, respectively. According to the description of the databases, the latter two databases contain a higher variability of speakers and consider more emotional classes than IEMOCAP, being more challenging scenarios to classify emotions. In addition, emotions share acoustic characteristics that make their discrimination more difficult. For these reasons, we believe that our feature sets are not sufficiently discriminatory to deal with the speaker and emotion variabilities contained in these databases.

For the case of CS evaluation, our approach based on articulation features yielded the best results with a good balance between sensitivity (76.0%) and specificity (72.4%). Although openSMILE was as accurate as articulation, its imbalance between SEN (82.9%) and SPE (62.8%) indicated a bias to recognize the dissatisfied class. The good performance of our feature set is based on the fact that customers under stress are prone to produce more hesitation and tense the muscles responsible for speech production. This, in turn, produces abnormal articulation patterns while they speak. Moreover, unlike the experiments with the emotional databases, the CS task was a binary classification task.

It is believed that the proposed approach is highly competitive considering for instance, the reduced number of features extracted with the articulation approach (488) compared to the 1584-dimensional feature vector extracted with the openSMILE toolkit. Additionally, the results when considering non-controlled acoustic conditions and non-acted opinions about a received service make us to think that the proposed approach is more convenient for industrial applications. Further work will explore the usability of the proposed method to estimate CS at call level.

6.3 Conclusion about multimodal analysis of customer satisfaction

This experiment addresses the automatic analysis of customer satisfaction using speech and text features extracted from the voicemails recordings of Konecta voicemails. Speech features were modeled by considering speaker embeddings, acoustic features, articulation features based on onset/offset transitions. Text features was modeled by word2vec representation of the text transcriptions generated by the hybrid-TDNN recognizer. Two types of fusion schemes were used: EF and GMU. The classification was performed using a SVM and a simple fully-connected DNN.

Unimodal analysis was also considered to set the baseline. The best performance was obtained by the word2vec representation with both classifiers. This result was expected since, from experience, there are keywords that well-represent each category. The satisfied customer frequently pronounced words related to gratitude and its vocabulary is small. There are even recordings where only "thanks" or "good service" are uttered. The unsatisfied customers tend to talk in detail about service- or business-related problems. This category commonly contains a vocabulary larger than the satisfied class. There are also keywords that reflect dissatisfaction such as "inconvenient", "bad", or "problem".

The initial hypothesis in multimodal analysis was that the CS classification could improve the performance when considering GMU to fuse the feature sets, since GMU can control the importance of each feature to model inner patterns (i.e, CS in this case). Overall, the EF-DNN and GMU approaches performed best when word2vec representation was presented. Although SVM was as accurate as the deep learning approaches, its training cost was higher due to its complex optimization process. When comparing the results between the unimodal and multimodal analyses, no significant differences were observed. However, it is believed that multimodal analysis is more robust than unimodal analysis, as each modality provides complementary information to better represent the pattern. This capability could be demonstrated after turning off one of the modalities. Since one of the information sources was lost, the classifier had to decide only based on the remaining mode. The experiment was performed using the open+w2v combination. The performance of the classifiers reached higher values in the ROC curve than the chance level (i.e., 50.0%) even when the most representative feature (w2v) was absent. This means that the classifier was able to model the pattern based on each of the modalities, with the combination being the best representation due to the complementary information from each source. On contrary, if the only modality in a unimodal approach was absent, the result is a decision by chance. This experiment also showed that GMU was superior to EF-DNN, supporting the initial hypothesis that GMU can better control the degree of importance of each model according to the operating conditions. Further work will explore the usability of the proposed method to estimate CS at call level. Future work will explore more complex textual representations such as BERT.

Conferences & Publications

The following publications emerged from the development of this master's thesis:

Journals

- **Parra-Gallego, L. F.**, & Orozco-Arroyave, J. R. (2021). Classification of emotions and evaluation of customer satisfaction from speech in real world acoustic environments. *Digital Signal Processing*, in press.

- Orozco-Arroyave, J. R., Arias-Vergara, T., Klumpp, P., Pérez-Toro, P. A., **Parra-Gallego, L. F.**, Roth, N., et al. (2020). Apkinson: the smartphone application for telemonitoring Parkinson's patients through speech, gait and hands movement. *Neurodegenerative Disease Management*, 10(3), 137-157.

Book chapters

- **Parra-Gallego, L. F.**, Arias-Vergara, T., & Orozco-Arroyave, J. R. (2021). Robust Automatic Speech Recognition for Call Center Applications. In *Workshop on Engineering Applications* (pp. 72-83). Springer, Cham.

Conferences

- Vásquez-Correa, J. C., Arias-Vergara, T., Klumpp, P., Strauss, M., Küderle, A., **Parra-Gallego, L. F.**, Roth, N., et al. (2019). Apkinson: A Mobile Solution for Multimodal Assessment of Patients with Parkinson's Disease. In *INTERSPEECH* (pp. 964-965).

List of Figures

Bibliography

[1] G. Zweig, O. Siohan, G. Saon, B. Ramabhadran, D. Povey, L. Mangu, and B. Kingsbury, "Automated quality monitoring in the call center with asr and maximum entropy," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, IEEE, vol. 1, 2006, pp. I–I.

[2] L. R. Rabiner and R. W. Schafer, *Introduction to Digital Speech Processing*. now Publishers Inc, 2007.

[3] S. Karpagavalli and E. Chandra, "A review on automatic speech recognition architecture and approaches," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 4, pp. 393–404, 2016.

[4] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. Springer, 2015.

[5] N. Shmyrev, *Quora dnn-hmm acoustic models better than gmm-hmm*, https://www.ameyo.com/blog/5-reasons-why-call-recording-is-imperative-for-your-contact-center, Accessed: 2019-09-05, 2017.

[6] T. Tanaka, R. Masumura, T. Moriya, T. Oba, and Y. Aono, "A joint end-to-end and dnn-hmm hybrid automatic speech recognition system with transferring sharable knowledge.," in *INTERSPEECH*, 2019, pp. 2210–2214.

[7] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015, pp. 5206–5210.

[8] K. Kinoshita, M. Delcroix, S. Gannot, E. A. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, *et al.*, "The reverb challenge: A benchmark task for reverberation-robust asr techniques," in *New Era for Robust Speech Recognition*, Springer, 2017, pp. 345–354.

[9] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, 2019.

[10] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 4845–4849.

[11] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*, Association for Computational Linguistics, 1992, pp. 357–362.

[12] P. Agrawal and S. Ganapathy, "Interpretable filter learning using soft self-attention for raw waveform speech recognition," *arXiv preprint arXiv:2001.07067*, 2020.

[13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, *Darpa timit acoustic phonetic continuous speech corpus cdrom*, 1993.

[14] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *arXiv preprint arXiv:1904.03416*, 2019.

[15] M. Ravanelli, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, and M. Omologo, "The dirha-english corpus and related tasks for distant-speech recognition in domestic environments," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2015, pp. 275–282.

[16] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth'chime'speech separation and recognition challenge: Dataset, task and baselines," *arXiv preprint arXiv:1803.10609*, 2018.

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint arXiv:1904.08779*, 2019.

[18] J. J. Godfrey, E. C. Holliman, and J. McDaniel, “Switchboard: Telephone speech corpus for research and development,” in *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 1, 1992, pp. 517–520.

[19] D. Haws and X. Cui, “Cyclegan bandwidth extension acoustic modeling for automatic speech recognition,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 6780–6784.

[20] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, “Improvements to deep convolutional neural networks for lvcsr,” in *2013 IEEE workshop on automatic speech recognition and understanding*, IEEE, 2013, pp. 315–320.

[21] K. Wang, N. An, B. N. Li, Y. Zhang, and L. Li, “Speech emotion recognition using fourier parameters,” *IEEE Transactions on Affective Computing*, vol. 6, pp. 69–75, 2015.

[22] N. Semwal, A. Kumar, and S. Narayanan, “Automatic speech emotion detection system using multi-domain acoustic feature selection and classification models,” in *Proceedings of the IEEE International Conference on Identity, Security and Behavior Analysis*, 2017, pp. 1–6.

[23] F. Eyben, K. Scherer, B. Schuller, J. Sundberg, E. André, C. Busso, L. Devillers, J. Epps, P. Laukka, S. Narayanan, *et al.*, “The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing,” *IEEE Transactions on Affective Computing*, vol. 7, pp. 190–202, 2015.

[24] R. Xia and Y. Liu, “Using i-vector space model for emotion recognition,” in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2012, pp. 2230–2233.

[25] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan, “Analysis of emotion recognition using facial expressions, speech and multimodal information,” in

Proceedings of the International Conference on Multimodal Interfaces, 2004, pp. 205–211.

[26] C. Busso, M. Bulut, C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. Chang, S. Lee, and S. Narayanan, "IEMOCAP: Interactive Emotional Dyadic Motion Capture Database," *Language Resources and Evaluation*, vol. 42, no. 4, p. 335, 2008.

[27] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. Narayanan, "The INTERSPEECH 2010 paralinguistic challenge," in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2010, pp. 2794–2797.

[28] H. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for speech emotion recognition," *Neural Networks*, vol. 92, pp. 60–68, 2017.

[29] R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, "X-vectors meet emotions: A study on dependencies between emotion and speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 7169–7173.

[30] D. Issa, M. Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomedical Signal Processing and Control*, vol. 59, 2020.

[31] M. Sajjad, S. Kwon, *et al.*, "Clustering-based speech emotion recognition by incorporating learned features and deep bilstm," *IEEE Access*, vol. 8, pp. 79 861–79 875, 2020.

[32] S. Roy, R. Mariappan, S. Dandapat, S. Srivastava, S. Galhotra, and B. Peddamuthu, "Qart: A system for real-time holistic quality assurance for contact center dialogues," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[33] A. Ando, R. Masumura, H. Kamiyama, S. Kobashikawa, Y. Aono, and T. Toda, "Customer satisfaction estimation in contact center calls based on a hierarchical multi-task model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 715–728, 2020.

[34] I. Siegert and K. Ohnemus, "A new dataset of telephone-based human-human call-center interaction with emotional evaluation.," in *ISCT*, 2015, pp. 143–148.

[35] J. Orozco-Arroyave, J. Vásquez-Correa, J. Vargas-Bonilla, R. Arora, N. Dehak, P. Nidadavolu, H. Christensen, F. Rudzicz, M. Yancheva, H. Chinaei, *et al.*, "Neurospeech: An open-source software for parkinson's speech analysis," *Digital Signal Processing*, vol. 77, pp. 207–221, 2018.

[36] S. Molau, M. Pitz, R. Schluter, and H. Ney, "Computing mel-frequency cepstral coefficients on the power spectrum," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, IEEE, vol. 1, 2001, pp. 73–76.

[37] J. Hui, *Ameyo speech recognition — feature extraction mfcc & plp*, https://medium.com/@jonathan_hui/speech-recognition-feature-extraction-mfcc-plp-5455f5a69dd9, Accessed: 2020-06-06, 2019.

[38] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[39] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, classifiaction," 1992.

[40] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[43] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth annual conference of the international speech communication association*, 2015.

[44] A. Waibel, "Modular construction of time-delay neural networks for speech recognition," *Neural computation*, vol. 1, no. 1, pp. 39–46, 1989.

[45] G. Mishne, D. Carmel, R. Hoory, A. Roytman, and A. Soffer, "Automatic analysis of call-center conversations," in *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2005, pp. 453–459.

[46] J. Arevalo, T. Solorio, M. Montes-y-Gómez, and F. A. González, "Gated multimodal units for information fusion," *arXiv preprint arXiv:1702.01992*, 2017.

[47] J. Thiemann, N. Ito, and E. Vincent, "Demand: A collection of multichannel recordings of acoustic noise in diverse environments," in *Proc. Meetings Acoust*, 2013.

[48] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth annual conference of the international speech communication association*, 2015.

[49] S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in north american english," *PloS one*, vol. 13, 2018.

[50] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *Ninth european conference on speech communication and technology*, 2005.

[51] E. McCrum-Gardner, "Which is the correct statistical test to use?" *British Journal of Oral and Maxillofacial Surgery*, vol. 46, no. 1, pp. 38–41, 2008.

[52] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks.," in *Interspeech*, 2018, pp. 3743–3747.

[53] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[54] M. Ravanelli, T. Parcollet, and Y. Bengio, "The pytorch-kaldi speech recognition toolkit," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 6465–6469.

[55] H. Schröter, T. Rosenkranz, A. Maier, *et al.*, "Clc: Complex linear coding for the dns 2020 challenge," *arXiv preprint arXiv:2006.13077*, 2020.

[56] Y. Park, S. Patwardhan, K. Visweswariah, and S. C. Gates, "An empirical analysis of word error rate and keyword error rate," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[57] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: A resource for the next generations of speech-to-text.," in *LREC*, vol. 4, 2004, pp. 69–71.

[58] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 2011.

[59] W. Campbell, D. Sturim, and D. Reynolds, "Support Vector Machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308–311, 2006.

[60] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal processing*, 2006, pp. 97–100.

[61] C. You, K. Lee, and H. Li, "An SVM kernel with GMM-supervector based on the Bhattacharyya distance for speaker recognition," *IEEE Signal Processing Letters*, vol. 16, no. 1, pp. 49–52, 2008.

[62] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support Vector Machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2009, pp. 1559–1562.

[63] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5329–5333.

[64] B. Fuller, Y. Horii, and D. Conner, "Validity and reliability of nonverbal voice measures as indicators of stressor-provoked anxiety," *Research in Nursing & Health*, vol. 15, pp. 379–389, 1992.

[65] X. Li, J. Tao, M. Johnson, J. Soltis, A. Savage, K. Leong, and J. Newman, "Stress and emotion classification using jitter and shimmer features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, pp. 1081–1084.

[66] J. Orozco-Arroyave, *Analysis of speech of people with parkinson's disease.* Logos-Verlag, 2016, vol. 41.

[67] H. Teager, "Some observations on oral air flow during phonation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, pp. 599–601, 1980.

[68] L. He, "Stress and emotion recognition in natural speech in the work and family environments," *PhD, Rmit University*, pp. 1–218, 2010.

[69] G. Zhou, J. Hansen, and J. Kaiser, "Nonlinear feature based classification of speech under stress," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 201–216, 2001.

[70] R. Kompe and R. Kompe, *Prosody in speech understanding systems.* Springer, 1997, vol. 1307.

[71] K. Rao, S. Koolagudi, and R. Vempada, "Emotion recognition from speech using global and local prosodic features," *International Journal of Speech Technology*, vol. 16, pp. 143–160, 2013.

[72] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[73] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[74] S. Parvandeh, H. Yeh, M. Paulus, and B. McKinney, "Consensus features nested cross-validation," *Bioinformatics*, vol. 36, pp. 3093–3098, 2020.

[75] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[76] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *arXiv preprint arXiv:1310.4546*, 2013.

[77] X. Rong, “Word2vec parameter learning explained,” *arXiv preprint arXiv:1411.2738*, 2014.

[78] S. Reese, G. Boleda Torrent, M. Cuadros Oller, L. Padró, and G. Rigau Claramunt, “Word-sense disambiguated multilingual wikipedia corpus,” in *7th International Conference on Language Resources and Evaluation (LREC)*, 2010.

[79] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, “Probing the information encoded in x-vectors,” in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2019, pp. 726–733.

[80] M. Jalal, E. Loweimi, R. Moore, and T. Hain, “Learning temporal clusters using capsule routing for speech emotion recognition.,” in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2019, pp. 1701–1705.