LEARNING SPEECH EMBEDDINGS FOR SPEAKER ADAPTATION AND SPEECH UNDERSTANDING

BY

LEDA SARI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Mark Hasegawa-Johnson, Chair
Associate Professor Paris Smaragdis
Professor Minh Do
Professor Heng Ji

ABSTRACT

In recent years, deep neural network models gained popularity as a modeling approach for many speech processing tasks including automatic speech recognition (ASR) and spoken language understanding (SLU). In this dissertation, there are two main goals. The first goal is to propose modeling approaches in order to learn speaker embeddings for speaker adaptation or to learn semantic speech embeddings. The second goal is to introduce training objectives that achieve fairness for the ASR and SLU problems. In the case of speaker adaptation, we introduce an auxiliary network to an ASR model and learn to simultaneously detect speaker changes and adapt to the speaker in an unsupervised way. We show that this joint model leads to lower error rates as compared to a two-step approach where the signal is segmented into single speaker regions and then fed into an adaptation model. We then reformulate the speaker adaptation problem from a counterfactual fairness point-of-view and introduce objective functions to match the ASR performance of the individuals in the dataset to that of their counterfactual counterparts. We show that we can achieve lower error rate in an ASR system while reducing the performance disparity between protected groups. In the second half of the dissertation, we focus on SLU and tackle two problems associated with SLU datasets. The first SLU problem is the lack of large speech corpora. To handle this issue, we propose to use available non-parallel text data so that we can leverage the information in text to guide learning of the speech embeddings. We show that this technique increases the intent classification accuracy as compared to a speech-only system. The second SLU problem is the label imbalance problem in the datasets, which is also related to fairness since a model trained on skewed data usually leads to biased results. To achieve fair SLU, we propose to maximize the F-measure instead of conventional cross-entropy minimization and show that it is possible to increase the number of classes with nonzero recall. In the last two chapters, we provide

additional discussions on the impact of these projects from both technical and social perspectives, propose directions for future research and summarize the findings.

To my parents, for their love and support.

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor Prof. Mark Hasegawa-Johnson for his mentoring, allowing me to investigate various problems during my PhD studies and always encouraging doing good research.

I would like to thank my doctoral committee members, Professors Minh Do, Paris Smaragdis and Heng Ji. Without their comments and questions during the preliminary and final exams, this dissertation would not be complete.

I would like give special thanks to Samuel Thomas of IBM T.J. Watson Research Center for our long-term collaboration on various projects. We also thank the IBM-UIUC collaboration initiative for providing us this chance. The collaboration would not have come into reality without approvals of former and current speech team leaders Michael Picheny and Brian Kingsbury.

During the course of my PhD, I had a chance to collaborate with other Beckman Institute affiliates. I would like to thank Prof. Dan Morrow, Renato Azevedo, Kevin Gu, and Tarek Sakakini for our collaboration on the HealthEdvisor project and demos at various events.

I would like to thank our current and former lab members including but not limited to Yijia Xu, Amit Das, Xuesong Yang, Junzhe Zhu, Kiran Ramnath, John Harvill, Heting Gao and Jialu Li. I also want to mention Prof. Chang D. Yoo of KAIST and his students for the collaboration on our fairness project.

I would also like to thank my friend from "south of Green Street", Helen Makhdoumian, for reminding me the power of "Hay Gin" (Armenian Woman) in various senses including the historical feminist magazine by Hayganuş Mark. Her emotional support was very important for me.

I would like to thank Katarzyna Borowiec for supporting me on my doctoral defense day. Even though we met late at a late stage, we became thesis writing and walking buddies and I am really grateful for her time.

Last but not least, I would like to thank my parents Berç and Jülyet Sarı for their continuous love and support at every stage.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ASAO	Adaptation by Speaker-Aware Offsets.
ASR	Automatic Speech Recognition.
BIC	Bayesian Information Criterion.
BLSTM	Bidirectional Long Short-Term Memory.
CER	Character Error Rate.
CNN	Convolutional Neural Network.
CRF	Conditional Random Field.
CTC	Connectionist Temporal Classification.
DNN	Deep Neural Network.
E2E	End-to-end.
FC	Fully-connected (layer).
FN	False Negative.
FP	False Positive.
GMM	Gaussian Mixture Model.
HCI	Human-Computer Interaction.
HMM	Hidden Markov Model.
LDA	Linear Discriminant Analysis.
LSTM	Long Short-Term Memory.
MFCC	Mel-Frequency Cepstral Coefficients.
MSE	Mean-Squared Error.

NLP	Natural Language Processing.
PCA	Principal Component Analysis.
ReLU	Rectified Linear Unit.
SLU	Spoken Language Understanding.
SVM	Support Vector Machine.
TN	True Negative.
TP	True Positive.
UBM	Universal Background Model.
WER	Word Error Rate.
WCCN	Within-Class Covariance Normalization.
fMLLR	Feature-space Maximum-Likelihood Linear Regression.

LIST OF SYMBOLS

$\mathbf{1}$	Indicator function.
$\mathcal{A}$	Domain of the protected attribute in fairness.
$\alpha_{i,j}$	Attention weight between the i-th and j-th segments.
α_t	Forward variable of CTC at time t.
β_t	Backward variable of CTC at time t.
γ	Posterior probability based on the CTC loss.
λ	Hyperparameter weighting a loss function.
ϕ	The input-output mapping function due to a neural network.
π	A path in the CTC computation.
ψ	Similarity kernel between speech segments.
σ^{-1}	Logit function.
A	Protected attribute in fairness.
b	Bias vector.
d	Distance measure.
diag	Diagonal matrix.
F_θ	Weighted F-measure with weight θ. Default $\theta = 1$.
$\tilde{F}$	Centralized first-order Baum-Welch statistic.
g	Binary ground truth label.
h^l	Hidden layer activation at layer l.
$\widehat{h}$	Normalized/adapted hidden layer activation.

I	Identity matrix.
i	i-vector.
$\mathcal{J}$	Loss function.
K	Similarity kernel for attention.
$\mathcal{L}$	Loss function.
l	Adapted layer number.
l_V	Speaker factors with respect to the space V.
M	Mean supervector in i-vector extraction.
M_L	Lower part of the neural model (below the adaptation layer).
M_U	Upper part of the neural model (above the adaptation layer).
m	Estimated mean vector.
μ	Mean vector.
N_c	Expected count of Gaussian component c.
o	Speaker-aware offset.
P	Probability.
p	Phone label.
q	Senone label.
$\tilde{S}$	Second-order Baum-Welch statistic.
Σ	Covariance matrix.
s	Speaker label.
s_t	Speaker embedding at time t.
T	Transformation (projection) layer.
T	Total duration.
u	Utterance.
V	Total variability matrix.
W	Weight matrix.
X	Input in the counterfactual framework.

x	Acoustic features.
x	Embedding vector corresponding to x.
y	Target label.
$\hat{Y}$	Outcome in the counterfactual framework.
$\hat{y}$	Softmax output.
z	Auxiliary network output (usually its penultimate layer output).

CHAPTER 1

INTRODUCTION

In the last decade, deep neural networks (DNNs) have gained popularity as the modeling approach for many applications including automatic speech recognition (ASR) [1] and end-to-end spoken language understanding (SLU) [2]. In supervised training schemes, the main idea is to represent the input to output mapping usually using a highly nonlinear function of intermediate representations generated at the hidden layers of a neural network. These representations are also called embeddings. Learning embeddings is a very broad topic, hence, it covers much more than the scope of this dissertation. In this work, we specifically focus on ASR and SLU tasks, and propose modeling and training techniques.

The increasing popularity of machine learning applications has raised concerns about their bias and fairness [3, 4, 5]. There are many causes of unfairness of machine learning applications including but not limited to dataset bias due to historical and societal reasons, measurement bias, algorithmic bias, evaluation bias, etc. [6]. As a machine learning application, ASR is also subject to fairness concerns. For example, there have been studies showing that there is usually a performance gap between male and female speakers [7, 8] as well as black and white speakers [9]. In an SLU application, the problem usually arises from having highly imbalanced datasets in which most of the data belongs to a single class and rare classes have only a few examples. Therefore, the second objective of this dissertation is to learn fair speech embeddings. We will achieve this goal by introducing fair training criteria for the ASR and SLU tasks.

In the case of ASR, our main task is to modify the model such that its performance is relatively robust to the changes due to varying speaker characteristics; i.e., our task is speaker adaptation. Most of the existing speaker adaptation methods assume that the input speech is pre-segmented into single-speaker regions. However, in the case of rapid speaker turns or on-

line applications, it might be necessary to process unsegmented speech files on-the-fly. Hence, we propose a speaker adaptation method to handle such cases. We also provide a new framework to approach the speaker adaptation problem which is inspired by the fairness literature. In this work, we try reducing the ASR performance gap between certain protected groups.

In the case of SLU, our focus is on end-to-end (E2E) speech-to-intent and speech-to-image object mapping without performing ASR explicitly. Since most SLU corpora have limited amounts of speech, we propose a method to leverage a non-parallel text corpus. These SLU corpora are also highly label-imbalanced which causes bias towards certain labels in the output. In order to make them fair to the minority classes, we propose an empirical F-measure optimization method to train E2E SLU systems.

The motivations behind working particularly on ASR and SLU tasks are the following:

- ASR is one of the most common applications of speech processing and with the introduction of smart devices, they started to become part of the daily life.
  - Among ASR problems, we particularly focus on speaker adaptation because its effect is immediately experienced by the end-user. For example, if we take a pre-trained English ASR system such as the ASpIRE model [10] and try to transcribe non-native female speech (such as a sentence uttered by the author), we easily see that the system does not produce the desired outcome. Considering the user satisfaction, we believe that speaker adaptation is an important problem in ASR.
- As for the SLU task, our ultimate goal is to achieve high quality human-machine verbal interaction which means that we need to teach machines how to understand speech so that it can take actions accordingly, such as answering a question by the user or turning off the lights at home.
  - SLU also entails many subproblems. Here we focus on modeling and training approaches. We focus on speech-to-dialog act, speech-to-intent and speech-to-object problems which can be simply named as utterance classification tasks rather than, for instance, slot-filling.

– We also restrict ourselves to E2E speech-to-concept models as we are trying to learn speech embeddings rather than perform the conventional two-step ASR + NLP approach. The reasons for avoiding the conventional approach will be discussed in Section 2.2.

The main contributions of this work can be summarized as follows:

1. A speaker adaptation model that can handle utterances with speaker change in the input speech signal,

2. A counterfactually fair algorithm to train a speaker adaptation model,

3. A multi-view approach for end-to-end SLU to make use of text data in speech-to-concept tasks,

4. A method to empirically optimize a neural network with respect to the F-measure so that we can prevent the model from neglecting the minority classes.

Table 1.1: Summary of the chapters in terms of tasks and proposals

Task	Model	Criterion
Speaker adaptation	An auxiliary network (Ch. 3)	Counterfactual posterior matching (Ch. 4)
E2E SLU	A multi-view network (Ch. 5)	Deep F-measure (Ch. 6)

As mentioned above, in this dissertation, we focus on two major speech tasks, namely, speaker adaptation for ASR and E2E SLU. For each of these machine learning tasks, we first propose a novel model to tackle certain problems of these tasks and then we propose a training criterion to achieve fairness in these tasks. This structure is summarized in Table 1.1. Chapter 3 will introduce an auxiliary network that performs speaker adaptation, then this model will be combined with a speaker attention mechanism to perform joint speaker change detection and speaker adaptation. We will show that even though we do not explicitly make use of the change point information during training, we can learn to detect speaker changes while reducing the ASR error rate. Chapter 4 will describe a fair training method inspired from the counterfactual fairness of [11]. Chapter 5 will switch to the E2E SLU problem

and propose a multi-view model that allows us to use non-parallel text data to improve speech-only SLU. We will show that using a large amount of text to pre-train a shared classifier improves the speech-only speech-to-concept classification performance. Chapter 6 will introduce an objective function that trains a DNN to maximize an approximate F-measure instead of accuracy. The newly proposed deep F-measure achieves accuracy comparable to that of standard cross-entropy based training while increasing the coverage; i.e., the number of classes with nonzero recall.

The rest of this dissertation is organized as follows: Chapter 2 will present a summary of related prior work on speaker adaptation, E2E SLU and fairness in machine learning. As shown in Table 1.1, Chapters 3-6 are the core chapters, each of which introduces the problem, proposes the model and provides experimental results. Chapter 7 discusses the findings and the general impact of this research as well as the directions for future work. Chapter 8 summarizes the contributions and concludes this dissertation.

CHAPTER 2

BACKGROUND

This chapter summarizes the prior work related to speaker adaptation and E2E SLU techniques developed in this dissertation. As we will propose a method for combining speaker change detection with speaker adaptation, we will also briefly summarize speaker change detection methods. In addition to these tasks, we will also focus on training fair models and will briefly review the fairness literature in machine learning.

2.1 Speaker Adaptation and Change Detection

2.1.1 Speaker Adaptation for ASR

Although DNNs have been successfully used in ASR systems, their performance is still affected by the variability inherent in speech. One of the main sources of variability is the mismatch between training and test speakers. Techniques proposed to alleviate this problem include using speaker-informed input features to the DNNs [12, 13], adapting the model structure [14, 15] and using auxiliary adaptation models or features [16, 17, 18, 19, 20, 21, 22, 23]. From a different perspective, adaptation methods can also be classified as supervised or unsupervised based on whether they use additional text or labels for the test data in addition to audio.

In input feature adaptation systems, features are normalized using a transform such as feature-space maximum likelihood linear regression (fMLLR) [24, 12] or the features are augmented with speaker specific features such as i-vectors [25, 13]. Other methods modify the speaker independent DNN model by introducing speaker adaptive layers [26]. For example, [27] investigates the use of learning an affine transform after long short-term memory (LSTM) activations at different layers of the network. Alternatively, the network

structure is kept the same but the weights are adapted based on speakers [14]. Recently, auxiliary feature or auxiliary network based adaptation methods have become more popular as these methods usually require little or no adaptation data [21]. Such approaches extract speaker invariant intermediate features by adversarial training [19, 20]. In these systems, the auxiliary network performs speaker classification whereas the main network performs phone/senone classification. Auxiliary feature based systems are usually developed using sequence summary vectors [28] and they are often applied only to the fully-connected (FC) layers. However, recently some methods are extended for the adaptation of the LSTM layers. For example, in [21], the sequence summary idea is applied in an encoder-decoder based end-to-end framework.

One method for speaker adaptation is to use speaker embeddings to augment the input features or intermediate activations of the original system. These embeddings can be i-vectors, summary vectors [28, 21] generated by an auxiliary network or a speaker vector read from a memory block [29, 30].

Another method for speaker adaptation is to provide speaker codes to a main network to adjust the weights of a layer. These speaker codes can be i-vectors or they can be learned discriminatively. A supervised way of learning speaker codes for speaker adaptation of DNN-HMM systems is proposed in [16, 18]. Using speaker codes, these techniques learn a bias for the sigmoid nonlinearities at the output of FC layers. Parametrization of nonlinearities is also proposed in [31] but their method adjusts the learned sigmoid or rectified linear unit (ReLU) layers without a speaker code.

A different version of using speaker codes is to learn an affine transformation for the LSTM activations. In [22], i-vectors are input to an auxiliary control network that computes a weight and a bias vector. Then, these i-vector dependent transformations are applied to the main network layers.

$$W_{\text{control}} = W_w i + b_w \tag{2.1}$$

$$b_{\text{control}} = W_b i + b_b \tag{2.2}$$

$$\hat{h} = \text{diag}(W_{\text{control}})h + b_{\text{control}}, \tag{2.3}$$

where W_w, b_w, W_b, b_b parametrize the adaptive weight and bias according to the i-vector i and h, $\hat{h}$ denote unadapted and adapted LSTM activations. Another extension of this idea is studied in [27] where they adapt bidirec-

tional LSTM (BLSTM) layers by having a separate affine transformation for the forward and backward LSTM activations. All of these methods are supervised, in the sense that they require a certain amount of data known to be the utterances of the same test speaker, although no reference text transcription is required.

Our earlier work on adaptation by speaker-aware offsets [23, 32] can also be grouped under the affine transformation category where speaker embeddings generated through an auxiliary network are used as bias vectors and subtracted from main network activations. As compared to [23, 32], in the current work, we investigate more general affine transformations. We also experiment with adding a nonlinearity to the transformation.

Given that i-vectors are commonly used in adaptation experiments, we use them for comparison to our proposed approaches. We discuss the estimation of these features in the following subsection.

i-vectors

Extraction of i-vectors [25] aims at modeling speaker and environment variability using a total variability matrix and a total factor. I-vectors are mainly used in speaker identification applications. In such systems, i-vectors are compared using cosine similarity to achieve matching with the known speakers. Although the extracted i-vectors are used to represent speakers, in [25] it is discussed that there are still channel variation effects between different recordings of the same speaker and therefore the i-vectors should be projected onto spaces where there is greater separation between the vectors of different speakers and smaller distances between the vectors of the same speaker. These projection techniques include within-class covariance normalization (WCCN) and linear discriminant analysis (LDA).

Consider the representation of an utterance as a supervector M, created by concatenating the mean vectors of a speaker dependent Gaussian mixture model (GMM). The idea of i-vector is to express the utterance supervector M as a linear combination of a vector in speaker and environment dependent space and a speaker and environment independent supervector which is usually determined by a universal background model (UBM) [25]. This UBM is a GMM trained on all available training data in a speaker independent manner and the supervector is constructed by concatenating the mean

vectors of the Gaussians in the mixture. If the UBM supervector is denoted by m, then the utterance supervector can be decomposed as

$$M = m + Vi, \tag{2.4}$$

where V is a low rank matrix characterizing the speaker and environment space and w has a $\mathcal{N}(0, I)$ prior distribution. The components of w are called total factors and the vector w is called identity vector or i-vector [25]. Extraction of i-vectors consists of estimating the matrix V and computing w. The total variability matrix V is the so-called eigenvoice matrix [33] which is part of the factor analysis of the vector. Let there be L speech feature vectors $\{x_1, x_2, \cdots, x_L\}$ each of which has dimension F and let the UBM GMM Ω have C components. The required statistics are

$$N_c(u) = \sum_{t=1}^{L} P(c|x_t, \Omega) \tag{2.5}$$

$$\tilde{F}_c(u) = \sum_{t=1}^{L} P(c|x_t, \Omega)(x_t - m_c) \tag{2.6}$$

$$\tilde{S}_c(u) = \operatorname{diag}\left(\sum_{t=1}^{L} P(c|x_t, \Omega)(x_t - m_c)(x_t - m_c)^{\mathrm{T}}\right) \tag{2.7}$$

$$N(u) = [N_1, N_2, \ldots, N_C]^{\mathrm{T}} \tag{2.8}$$

$$F(u) = [\tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_C]^{\mathrm{T}} \tag{2.9}$$

$$S(u) = \begin{bmatrix} \tilde{S}_1 & & \\ & \ddots & \\ & & \tilde{S}_C \end{bmatrix}, \tag{2.10}$$

where c is the Gaussian component index, u is the utterance index, $P(c|x_t, \Omega)$ is the posterior probability of the Gaussian component c given x_t and the model parameters Ω, and superscript T denotes transposition. Equation (2.6) is the centralized version of the mean estimation formula in Baum-Welch training of GMM and m_c is the mean vector of the UBM mixture component c. Using these Baum-Welch statistics, inverse covariance of the speaker factors $l_V(u)$ and other statistics which are based on accumulation over all

utterances are computed:

$$l_V(u) = I + V^{\mathrm{T}}\Sigma^{-1}N(u)V \tag{2.11}$$

$$A_c = \sum_u N_c(u) l_V^{-1}(u) \tag{2.12}$$

$$D = \sum_u F(u) \left(l_V^{-1}(u) V^{\mathrm{T}}\Sigma^{-1}F(u)\right)^{\mathrm{T}} = [D_1, \ldots, D_C]^{\mathrm{T}}, \tag{2.13}$$

where Σ^{-1} is the inverse of the UBM covariance matrix. Estimation of V starts with an initial guess and then Eqs. (2.11)-(2.14) are used to update the value of V for a certain number of iterations.

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_C \end{bmatrix} = \begin{bmatrix} A_1^{-1}D_1 \\ \vdots \\ A_C^{-1}D_C \end{bmatrix}. \tag{2.14}$$

Once V is computed, the i-vector i is obtained by using Baum-Welch statistics which have similarities with maximum likelihood estimation of GMMs. As in the estimation of V, $N_c(u)$ and $\tilde{F}_c(u)$ are computed first and then the i-vector i is computed by

$$i = (I + V^{\mathrm{T}}\Sigma^{-1}N(u)V)^{-1}V^{\mathrm{T}}\Sigma^{-1}\tilde{F}(u). \tag{2.15}$$

Here, $N(u)$ is a $CF \times CF$ dimensional diagonal matrix with the diagonals $N_c I$ where I is the identity matrix of dimension F, and $\tilde{F}(u)$ is concatenation of $\tilde{F}_c(u)$. Σ is a diagonal covariance matrix estimated by the factor analysis which is described in [33].

x-vectors

In recent years, neural network based speaker embeddings have started to outperform i-vectors in speaker recognition. Hence they have also become popular for speaker adaptation. One of the most widely used network-based speaker embedding is the x-vector [34, 35]. The main idea in this case is to train a speaker classifier on a dataset with large number of speakers and compute statistics over intermediate layer activations to get the speaker embeddings. In particular, in [34], a time delayed neural network followed by a statistics pooling layer, which computes the mean and standard deviation of

its inputs and concatenates them, and two FC layers are used to construct a speaker classifier. The network is trained with cross-entropy objective and the embeddings are computed by taking the activations from either of the FC layers. This embedding is an utterance-level embedding instead of a frame-level one.

2.1.2 End-to-end ASR and CTC Loss

In recent years, the availability of very large speech corpora and increased computation power has led speech researchers to investigate E2E approaches for ASR. The main goal of these systems is to map the speech signal or acoustic features into characters or words directly, which eliminates the need for linguistic knowledge such as the lexicon. There are several paradigms for end-to-end ASR, including CTC-based models [36], RNN transducers [37], purely attention-based transducers [38], and joint models that combine CTC with encoder-decoder models [39]. In this study, we will focus on CTC-based models, which we will review next.

Neural network training using CTC loss [40], which was proposed for sequence-to-sequence labeling tasks, has become one of the major approaches for end-to-end ASR systems [36, 41]. For a given acoustic feature sequence $\mathbf{x} = [x_1, x_2, \ldots, x_T]$ where T is the total duration and an output sequence, in our case a character sequence, $\mathbf{c} = [c_1, c_2, \ldots, c_L]$, where L is the sequence length which is shorter than the input sequence, i.e. $L \leq T$, the goal is to write the probability of the output sequence given the input sequence. Since the sequences are usually of different lengths, the probability is decomposed into possible alignment paths π between input and output. Suppose that a neural network generates per-frame softmax outputs $\mathbf{y} = [y_1, y_2, \ldots, y_T]$ for the input $\mathbf{x}$, then

$$P(\mathbf{c}|\mathbf{x}) = \sum_{\pi:l(\pi)=\mathbf{c}} P(\pi|x) \tag{2.16}$$

$$= \sum_{\pi:l(\pi)=\mathbf{c}} \prod_{t} P(y_t(\pi_t)|\mathbf{x}). \tag{2.17}$$

The network ϕ that generates the softmax outputs $\mathbf{y} = \phi(\mathbf{x})$ is trained by

minimizing the negative log-likelihood, denoted by $\mathcal{L}_{\text{CTC}}$, as

$$\mathcal{L}_{\text{CTC}}(\mathbf{x}, \mathbf{c}) = -\operatorname*{logsumexp}_{\pi:l(\pi)=\mathbf{c}} \sum_t \log P(y_t(\pi_t)|\mathbf{x}). \tag{2.18}$$

As shown in [40], these probabilities can be computed using a forward-backward algorithm. The algorithm first starts with augmenting the original sequence with a special blank symbol $(-)$ and produces a new augmented sequence $\mathbf{l}' = [-, c_1, -, c_2, -, \ldots, -, c_L, -]$ of length $2L + 1$; then using the forward and backward variables α and β, the total probability of the observed sequence $\mathbf{c}$ can be written using any t as

$$P(\mathbf{c}|\mathbf{x}) = \sum_{s\in\{1,\ldots,2L+1\}} \alpha_t(s)\beta_t(s), \tag{2.19}$$

or specifically for the last time index T as

$$P(\mathbf{c}|\mathbf{x}) = \alpha_T(|\mathbf{l}'| - 1) + \alpha_T(|\mathbf{l}'|), \tag{2.20}$$

where $|\mathbf{l}'| = 2L + 1$ denotes the length of $\mathbf{l}'$.

Even though E2E systems are usually trained on very large corpora and considered to be more robust to speaker variability, there are studies which address the speaker adaptation problem for E2E ASR systems. These ideas either originate from speaker adaptation of conventional systems such as feature normalization or appending i-vectors to the inputs [42, 43], adversarial training approaches [44], or memory based architectures [29].

2.1.3 Speaker Change Detection

Speaker change detection is the task of finding the time instances in audio recordings when a different speaker starts to speak. One general approach to this problem is to use a distance-based method. These methods extract features using sliding windows, compare feature representations of consecutive windows using a distance measure and then threshold the distance [45].

On the other hand, model-based approaches fit a model to the features of individual segments and their concatenation, and choose the hypothesis with a higher score; this score can be the Bayesian information criterion (BIC) [46] or Gaussian likelihood score [47].

Among the most commonly used features to represent speaker characteristics of a speech segment are i-vectors [25]. Although i-vectors have been successfully used in speaker verification applications, reliability of these vectors depends on segment duration [48, 49]. For shorter segments, it is harder to estimate the i-vectors. In order to solve this problem, short speech segments are often clustered using BIC, Gaussian divergence [50] or x-means [51, 52] prior to computing the i-vector. However, these clustering methods are mainly designed for offline processing and cannot be used in low-latency applications [51].

Recently, neural network based speaker embeddings have been used as an alternative [53, 54] or as complementary features [34] to i-vectors. Studies have shown that network based embeddings can achieve better performance than using BIC based approaches on mel-frequency cepstral coefficients (MFCCs) [53, 54] or filterbank coefficients [55]. In [34], network embeddings are used in a speaker classification task with a probabilistic linear discriminant analysis backend and have been shown to achieve better performance than i-vectors especially when the inputs are short (<10 s). These embedding networks are trained using multiclass cross-entropy for speaker classification using a large number of speakers [34, 55], using contrastive loss on two inputs processed in a Siamese architecture [56] or using triplet loss [53]. In order to map variable length sequences to fixed dimensional embeddings, LSTM [57] layers are usually employed.

In addition to generating embeddings, neural networks have also been used in E2E speaker change detection systems [54, 58, 59, 60] where the change decision is made at the end of a network instead of thresholding a distance measure. These systems can be classified into cases where the problem is reduced to taking two speech segments as input and comparing them [54, 58] or deciding if there is a change point within a given single speech segment [59, 60]. The networks that compare two segments usually have a Siamese structure where the initial few layers processing the two inputs share their weights. A similar structure is also usually used in embedding generating networks where the training objective consists of comparing the features extracted from the shared Siamese layers.

2.2 Spoken Language Understanding

The goal of SLU is to extract the meaning of the spoken content. The "meaning" could be the underlying intent of the speaker, the speech act, the object described in a sentence or the goal could be to achieve slot-filling. Conventional SLU systems first convert the speech signal into text using an ASR system and then the text is processed by an NLP system to get the "meaning". Early systems usually extract lexical, prosodic features or word n-grams and use statistical modeling techniques such as hidden Markov models [61] to classify the speech features. Alternatively, CRFs [62] or SVMs [63] are used to classify the representations obtained from ASR output lattices. There are also NLP studies which assume that an ASR system already generated the text and then the NLP model focuses on text-based SLU. These works are usually based on classifying word representations. For example, in [64], 1-hot vectors or embeddings such as word2vec [65] are used for SLU. Recently, more powerful embeddings such as BERT embeddings [66] are used for joint intent classification and slot-filling [67].

One advantage of the two-step approach is that ASR and NLP components can be trained separately on different datasets. However, there are several disadvantages of this two-step approach:

1. Separately optimized ASR and NLP models may not give the optimal solution for the end-to-end problem, i.e. speech-to-concept (intent, slots, entities, etc.).

2. There is error propagation in the cascaded system. ASR output will have errors which will result in noisy text input to the NLP component which is usually trained on clean text data.

3. For some languages, it may not be possible to train an ASR system, necessitating methods directly applicable to speech signals.

4. Speech carries additional information such as pitch, prosody, etc., that reveals the emotion of the speaker that could help identify the intent better. Since text modality lacks these additional cues, text based SLU systems cannot make use of these cues.

In order to mitigate these disadvantages of the two-step approach, E2E SLU approaches have become popular in recent years.

E2E approaches for SLU include [2, 68, 69, 70]. Most of these approaches require large amounts of labeled speech data to achieve good performance. In [68], the authors attempt to predict intent labels directly from log-mel features. Although the speech-only accuracy is lower than a cascaded ASR+SLU system performance, the ASR+SLU degrades when tested with ASR based text. In [71], the authors aim at finding compact speech representations instead of using acoustic features directly to improve speech-only SLU. An encoder-decoder framework is used in [2], where the decoder is conditioned on the audio transcript. The authors conclude that having an intermediate text representation yields better performance than simply classifying acoustic features without any constraint. In our experiments, we also make similar observations and therefore use a text-based classifier pretraining to guide a subsequent speech-only training.

None of the mentioned studies tackle the problem of having non-parallel text and speech. In [69], an E2E approach for slot filling is introduced and the authors apply transfer learning starting from word recognition and going to named entity recognition and then slot filling to deal with data scarcity problem. However, they still require speech and the corresponding text to train the initial model and also labeled data for additional tasks. Another transfer learning approach is used in [70] where the authors first train a word recognizer and use it as a feature extractor or fine-tune those layers on the slot-filling task. Although the word recognizer and the SLU classifier can be trained on different datasets, the recognition system still requires large amounts of parallel speech and text. In [72], a cascaded approach is used where grapheme posteriors are generated from speech features and then the posterior features are classified. Although the graphemic part can be separately trained on an ASR corpus, and the SLU part on a text based dataset, this model still requires large amounts of parallel data.

In the SLU task, we focus on three application areas, namely, dialog act recognition, intent classification and speech-to-image label. In dialog act recognition, each utterance represents a speech act such as appreciation, disagreement, w-question. These acts are related to the speech acts of [73] or the illocutionary forces of [74]. On the other hand, intent classification aims at finding the effect that the speaker wants to convey to the listener [75, 76]. The third task is a dataset dependent problem. Consider a spoken image captioning dataset where each image comes with a spoken caption.

Then the image label which usually corresponds to the main object in the image becomes the theme of the spoken content. Our goal is to identify the image label given the spoken caption. We apply this to the SpeechCOCO dataset [77].

2.3 Fairness in Machine Learning

It has been observed that machine learning models are liable to making unfair decisions [78] due to various reasons including data bias, missing data from certain groups and algorithmic bias. To evaluate the unfairness in machine learning, several fairness criteria have been defined. An overview of some of these criteria is provided in [79]; several that are most relevant to this work will be described below after presenting a general classification of these objectives.

The problem of how to define fairness in machine learning was perhaps first considered by Pearl [80], who published causal machine learning models based on an earlier statistical analysis of college admissions data [81]. In this study, there was an explicit comparison between a broad definition of fairness, comparable to what is currently called group fairness such as demographic parity [82], versus the modern concept of individual fairness [83].

In the last decade, several interpretations of fairness have led to various definitions which are usually grouped into two major categories:

1. Group fairness measures: Demographic parity [83], equalized odds [84], equal opportunity [84], and conditional statistical parity [85] are among these measures which aim at treating different groups equally.

2. Individual fairness measures: Fairness through awareness [83], fairness through unawareness (blindness) [86] and counterfactual fairness [11] are among these measures which aim at producing similar outcomes for similar individuals.

There are theoretical results showing that some of these constraints cannot be achieved simultaneously, which is also called the "impossibility theorem of fairness" [87, 88]. It states that "no more than one of the three fairness metrics of demographic parity, predictive parity and equalized odds can hold at the same time for a well calibrated classifier" [88]. There is also a debate

on whether there is a conflict between individual and group fairness [89]. Another source of debate is whether these constraints must inherently lead to a reduction in the original performance criterion such as the accuracy of a model [90]. In our work, we will focus on individualized versions of equalized odds and equal opportunity, and also counterfactual fairness which we will review next.

According to [84], **equalized odds** is defined as the condition in which a predictor $\hat{Y}$ of an outcome Y is conditionally independent of the sensitive attribute A given Y. Although it is applicable to the cases where Y belongs to a binary, multi-class, or continuous spaces, especially for the case where the outcome and the attribute are binary, equalized odds reduce to the following: For $y \in \{0, 1\}$,

$$P(\hat{Y}=1|A=0, Y=y) = P(\hat{Y}=1|A=1, Y=y). \tag{2.21}$$

In [84], **equal opportunity** for a binary predictor is defined as

$$P(\hat{Y}=1|A=0, Y=1) = P(\hat{Y}=1|A=1, Y=1). \tag{2.22}$$

Here, the outcome $\hat{Y} = 1$ is defined to be "advantaged," and hence Eq. (2.22) "requires non-discrimination only within the 'advantaged' outcome group." Since this definition only requires matching of true positives instead of the outcome distribution, it is a weaker criterion as compared to the equalized odds.

Equalized odds and equal opportunity can be classified as group fairness criteria [83], in that they measure fairness by comparing outcome probabilities aggregated across all members of a group. In contrast, **individual fairness** criteria seek to enforce similar treatment of similar individuals, by specifying that the difference in outcomes must be smaller than the difference between individuals. Suppose that $u, v \in \mathcal{U}$ are individuals, and that the classifier M is defined as the mapping $M : \mathcal{U} \to \mathcal{M}$, where $\mathcal{M}$ is the set of distributions over outcomes. Assume the existence of metrics $d : \mathcal{U} \times \mathcal{U} \to \Re$ and $D : \mathcal{M} \times \mathcal{M} \to \Re$; according to [83], individual fairness requires that

$$D(Mu, Mv) \leq d(u, v). \tag{2.23}$$

In [11], the notion of **counterfactual fairness** is introduced which states

that if we intervene with a sensitive attribute $A \in \mathcal{A}$, the probability of outcomes should match. More formally, for $a, a' \in \mathcal{A}$,

$$P(\hat{Y}_{A \leftarrow a} | X = x, A = a) = P(\hat{Y}_{A \leftarrow a'} | X = x, A = a), \tag{2.24}$$

where $A \leftarrow a$ shows the intervention that A is assigned to a according to *do-calculus* [91] while predicting $\hat{Y}$, and a' is the counterfactual value which is also an element in the domain of the sensitive attribute A, i.e. $\mathcal{A}$. In other words, counterfactual fairness states that irrespective of the sensitive attribute such as gender, race, etc., and the ground truth value of the outcome, the predicted outcome probabilities should match.

Given a causal graph explaining a problem, the three main steps of counterfactual inference are abduction, action and prediction. The abduction and prediction steps require separation of the attributes, X, into those that are causally dependent on A (the descendants, X_d) and those with no such causal dependence (the non-descendants, X_n, possibly including unobserved latent variables). In the abduction step, given the prior of the latent variables and the observations, the posterior of the latent variables are computed. In the action step, intervention is applied. In the prediction step, the resulting distribution for the variables except the intervened and latent variables is computed using the results from first two steps.

Counterfactual fairness may be defined to be either a group-fairness or an individual-fairness criterion [92], depending on whether or not the metric $d(u, v)$ in Eq. (2.23) considers any of the descendant attributes, X_d. In the context of speaker adaptation, for example, consider pitch. A speaker's observed pitch frequency, X_d, is causally dependent on gender, but is also causally dependent on an unobservable set of latent variables X_n including, for example, the speaker's vocalis mass relative to others of the same gender, the speaker's habitual prosody, and the speaker's prosody in the observed utterance. A metric $d(u, v)$ that measures differences in X_d will rarely consider men and women to be similar, but a metric measuring differences in X_n might more frequently find men and women to be comparable.

Recently, there have been studies in computer vision [93, 94] and natural language processing [95, 96] which use the abducted distribution of the residual variable, together with counterfactually modified sensitive attributes, to generate a counterfactual dataset, which is then used to train the model.

However, to our knowledge, our study is the first proposal for counterfactual training for ASR.

Individual fairness is appropriate in speech recognition because speech signals are highly speaker dependent. Hence, achieving similar group level error rates does not guarantee individual-level performance due to within-class variability of individuals. For example, two female speakers belonging to the same race can have highly different pitch; therefore, we cannot treat these two speakers in the same way in terms of ASR.

At the same time, individual fairness at the level of descendant attributes (e.g., absolute pitch) is uninteresting, because it is the *status quo:* identical speech signals already produce identical outcomes. It is more interesting to consider **individualized counterfactual fairness**, i.e., individual fairness with respect to the non-descendants of A. As in recent papers in natural language processing [95] and computer vision [94], similarity of latent attributes ($d(u, v)$) is judged by counterfactual signal generation, and is used to enforce similarity of classifier outcomes ($D(Mu, Mv)$).

The criteria mentioned above are mainly concerned with accuracy rates, precision or the error rates which are usually defined through true positives TP, false positives FP, true negatives TN and false negatives FN. However, there are other measures which combine these statistics; e.g., the **F-measure** takes the harmonic mean of precision and recall. As will be discussed in Chapter 6, in our SLU problems, the datasets are highly imbalanced. For instance, the ATIS dataset has roughly 75% "flight" intent which means that if we are not careful about modeling, we can easily output one label all the time and achieve 75% accuracy, but under this condition, F-measure will be quite low. Hence, as a new fairness objective, we propose to maximize F-measure to achieve fairness in the E2E SLU problem for imbalanced datasets.

First, consider the binary classification problem. Given the true positive (TP), false positive (FP) and false negative (FN) counts for a test dataset, precision (Prec) and recall (Rec) of the model can be written as follows:

$$\text{Prec} = \frac{TP}{TP + FP} \quad \text{and} \quad \text{Rec} = \frac{TP}{TP + FN}. \tag{2.25}$$

Given these definitions, F_θ measure is defined as a weighted harmonic mean

of precision and recall [97]

$$F_\theta = \frac{(1+\theta^2)\text{Prec} \cdot \text{Rec}}{\theta^2 \text{Prec} + \text{Rec}}. \tag{2.26}$$

If we substitute the precision and recall expressions to the above equation, we can also write the F_θ measure as

$$F_\theta = \frac{(1+\theta^2)TP}{\theta^2(TP+FN)+(TP+FP)}. \tag{2.27}$$

For the multi-class classification case, there are several ways of computing the F_θ-measure. We can compute the average precision and recall over all classes and then take their harmonic mean to get the micro-F_θ-measure. Alternatively, we compute the class-wise F_θ-measures and take the average over classes to get the average-F_θ-measure. In this work, we optimize the latter. Suppose that there are K classes and N_k denotes the number of data points from class k, then the average F_θ is computed as

$$F_\theta = \frac{1}{K}\sum_{k=1}^{K} \frac{(1+\theta^2)TP(k)}{\theta^2 N_k + (TP(k)+FP(k))}. \tag{2.28}$$

Note that the N_k term corresponds to $(TP(k)+FN(k))$.

There have been several studies on F-measure maximization [98, 99, 100, 101, 102, 103]. These models usually focus on binary classification using non-neural-network models: a situation in which the problem of F-measure optimization reduces to the problem of learning a threshold on the scores computed by the model to make a decision. We are aware of one study [102] that performs F-measure optimization for convolutional neural networks, but again, using a system that generates several binary classification outputs in parallel; in this scenario, F-measure optimization reduces to the task of tuning the thresholds of individual binary classifiers in order to maximize a weighted log likelihood. However, true multi-class classification, using the softmax output of the neural network, requires a modified definition of the F-measure. There is no threshold that can be tuned; instead, F-measure optimization requires optimizing the model itself to generate "better" scores in terms of the F-measure. Model versus threshold optimization is the fundamental difference between our study and the previous ones.

CHAPTER 3

SPEAKER EMBEDDINGS FOR SPEAKER ADAPTATION

In this chapter, we will first review our adaptation method proposed in [23, 32] and then extend its application to the cases where a speaker change occurs in the input utterance as it appears in [104]. The basic idea is to combine the Siamese network idea with the auxiliary network using a speaker attention mechanism. We will present our results in terms of ASR and speaker change detection performance. We will show that although we do not explicitly use the speaker change information during training, the model learns to detect the speaker changes.

3.1 Introduction

As in many machine learning applications, ASR performance degrades on unseen data, especially on inputs from unseen speakers. This is largely due to the significant acoustic variations found in speech signals produced by different individuals even when they speak the same words. Physical differences between individuals such as vocal tract length, and idiolectal differences such as region and social grouping, affect the way we speak. These factors contribute to changes in prosody and segmental articulation along with other variations. To alleviate this problem, several methods have been proposed as discussed in Chapter 2.

Most of the existing speaker adaptation systems assume that the input utterances are pre-segmented into single-speaker regions and adaptation is usually applied to these regions. However, in an online application or in cases where there are rapid speaker changes such as in dialogues, this two-step approach will take longer than having a single joint network that detects speaker changes and adapts simultaneously. The online scheme also requires an unsupervised speaker adaptation as we do not have access to speaker

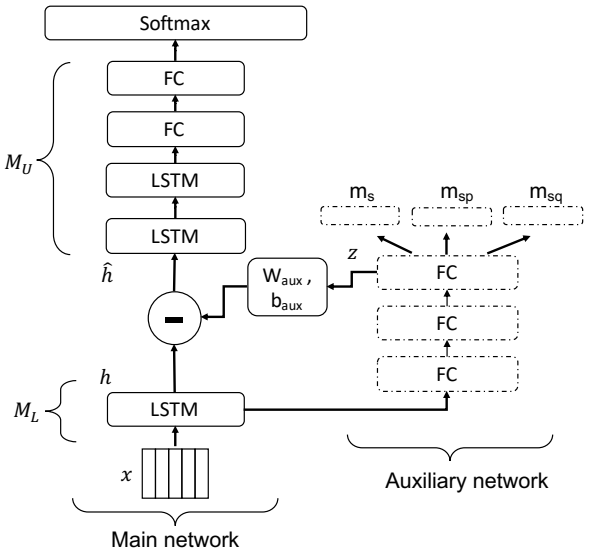


Figure 3.1: Flowchart of the adaptation by speaker-aware offsets method of [32]

identity during test time.

In this chapter, the problem we address is whether we can construct a single acoustic model that can detect the speaker change and automatically adapt to different speakers simultaneously. In order to develop such a system, we combine ideas from neural network based speaker change detection [105] and unsupervised speaker adaptation by an auxiliary network [23, 32]. The key novel contribution of this work is a method that moves the speaker-change decision inside the speech recognition system, in the form of a soft-decision speaker-attention layer. The method then trains the speaker-attention layer explicitly in order to minimize the ASR error rate. The speaker-attention layer is used to accumulate a soft-decision speaker embedding, and from that point onward, the network behaves similarly to [32]. In addition to the mean normalization proposed in [32], we also investigate an affine and a nonlinear transformation of these activations that depend on the speaker embedding generated by the auxiliary network. We also show that the learned speaker embeddings can be used for speaker segmentation although we do not explicitly train the network with this objective.

3.2 An Auxiliary Network for Speaker Adaptation

The current auxiliary network is an extension of our previous work on auxiliary network based speaker adaptation [23, 32] which performs speaker dependent mean removal from the main network activations. As shown in Fig. 3.1, the architecture consists of a main network that performs senone classification for ASR. An auxiliary network consisting of FC layers is then attached at the output of one of the LSTM layers of the main network which splits the network into the lower M_L and upper M_U parts. The auxiliary network in the previous studies is trained to reconstruct the speaker, (speaker, phone) and (speaker, senone)-level averages of the LSTM outputs (m_s, m_{sp}, and m_{sq} in Fig. 3.1). The last FC layer before the output layers from the auxiliary network is used to extract a speaker-aware vector and then this vector is transformed by an affine layer to get speaker-aware offsets. This corresponds to adapting the bias of the LSTM output depending on auxiliary network predictions of the speaker, phone, and senone. Let h denote the LSTM activations from the last LSTM in the lower part of the main network, and z denote the last FC layer output of the auxiliary network, then the speaker-aware offset o_{aux} and the normalized (adapted) LSTM activations $\hat{h}$ are determined by

$$o_{\text{aux}} = W_{\text{aux}} z + b_{\text{aux}} \tag{3.1}$$

$$\hat{h} = h - o_{\text{aux}}. \tag{3.2}$$

It has been shown that this method achieves better performance than adapting the input features using fMLLR. However, it assumes that inputs are pre-segmented by speaker and cannot handle utterances with change points in them. Hence, it requires pre-processing by an additional online speaker change detection module.

3.3 Joint Speaker Adaptation and Speaker Change Detection

The proposed model shown in Fig. 3.2 extends the auxiliary network described above by introducing a speaker attention mechanism. Thus, it achieves

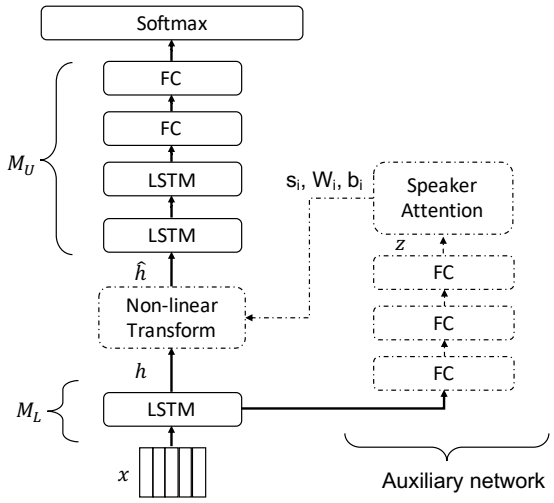


Figure 3.2: Architecture of the joint network consisting of a main and an auxiliary network

simultaneous ASR and speaker change detection. In the proposed model, similar to Fig. 3.1, we split the main network into M_L and M_U at the end of l-th LSTM layer ($l = 1$ in the figure) at which point we attach the auxiliary network. On the other hand, as compared to Fig. 3.1, in Fig. 3.2, the auxiliary network does not have any output layers. Instead, it has a speaker attention layer that produces the parameters for segment level transformation. Since we do not have the auxiliary outputs in the system, during training, we only backpropagate the senone classification cross-entropy loss and we do not apply multitask objective based learning in the new model as compared to the model described in Section 3.2.

The goal of the similarity based speaker attention layer introduced here is to emulate the distance based comparison of segment embeddings in a Siamese network where the embeddings are produced by a shared network. In our case, M_L and the auxiliary layers act as these shared layers. Instead of making hard change decisions, we leave them soft and compute speaker-aware vectors which are in turn used to determine the transformation for the main network activations in a nonlinear fashion as we describe next.

Consider an input acoustic feature sequence $[x_1, x_2, \ldots, x_T]$ processed by l LSTM layers which generate activations $[h_1^l, h_2^l, \ldots, h_T^l]$ at the end. We then pass these activations to the auxiliary network. The auxiliary network transforms its inputs using FC layers followed by nonlinearity, and after its final FC layer, it applies average pooling over time with a window of length

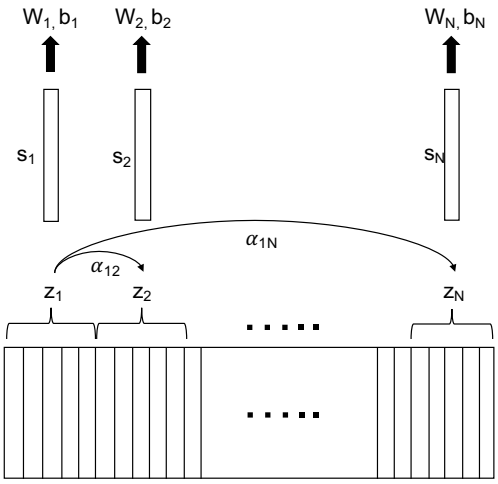


Figure 3.3: Steps for computing the segment dependent weight and bias

T_w and shift T_s, here we take $T_s = T_w$. This operation gives us representative embeddings for the segment corresponding to each window.

Suppose that $[z_1, \ldots, z_N]$ denote the average activations at the end of pooling for each window where $N = \frac{T - T_w + T_s}{T_s}$ denotes the total number of windows in the utterance. Then, soft speaker change detection is performed by an attention mechanism:

$$\psi(z_i, z_j) = \mathrm{ReLU}(z_i \cdot z_j) \tag{3.3}$$

$$\alpha_{i,j} = \frac{\psi(z_i, z_j)}{\sum_{j'} \psi(z_i, z_{j'})}. \tag{3.4}$$

Instead of Eq. (3.4), softmax type of normalization can also be used. However, our experiments showed that the version in Eq. (3.4) performs better. The speaker code for the i-th subsegment, s_i, is then computed by

$$s_i = \sum_{j=1}^{N} \alpha_{i,j} z_j. \tag{3.5}$$

These steps are visualized in Fig. 3.3. After frame-level activations have been computed at the end of the auxiliary network, average-pooled activations z_i are estimated corresponding to each segment. Speaker embeddings s_i are then estimated based on soft alignment α values. Using this speaker code s_i, we apply either an affine (either bias-only or weight and bias) or a nonlinear transformation to the main network activations h. In the first case, we use

s_i as speaker offsets and remove them from the main network activations as in [32]:

$$\hat{h}_t = h_t - s_i, \quad t \in [iT_s, (i+1)T_s). \tag{3.6}$$

In the second case, we apply an affine transformation where the weight and bias are parametrized by W_w, b_w, W_b and b_b. These give us the segment-specific weight and bias W_i, b_i (also shown in Fig. 3.3).

$$W_i = W_w s_i + b_w \tag{3.7}$$

$$b_i = W_b s_i + b_b. \tag{3.8}$$

We then apply the segment-dependent affine transformation to h in the following way:

$$\hat{h}_t = \text{diag}(W_i) h_t + b_i, \quad t \in [iT_s, (i+1)T_s). \tag{3.9}$$

In the third case, we apply a nonlinearity after the affine transformation, thus the relation between h and $\hat{h}$ becomes the following:

$$\hat{h}_t = \tanh(\text{diag}(W_i) h_t + b_i), \quad t \in [iT_s, (i+1)T_s). \tag{3.10}$$

Here the parameters to be learned are the parameters of the auxiliary network that generate the z_i's, and also W_w, b_w, W_b, and b_b. As will be shown in the experiments, we found that nonlinear transform is the most effective one among these three options.

Training of the model starts with training the main network without the auxiliary one using cross-entropy as the objective function. We then train the parameters of the auxiliary network by either freezing the main network or fine-tuning the M_L part of the network. This second phase training is performed on utterances with change point in them. Our training objective for training the auxiliary network is still the cross-entropy from the main network without an additional multitask loss.

As will be shown in the next section, the proposed model achieves not only unsupervised speaker adaptation but also online speaker change detection, thanks to the speaker attention layer, even though it is not explicitly trained to detect them. Therefore, this joint model combines and extends the two

Table 3.1: Amount of training, heldout and test data for main network training for BN and SWB datasets

	BN		SWB	
	Speakers	Duration (hr)	Speakers	Duration (hr)
Train	2201	104.1	519	257.0
Heldout	275	19.2	90	5.0
Test	275	11.8	40	2.1

separate systems described above.

3.4 Experiments

Experiments are performed on the Broadcast News (HUB4, BN) [106, 107] and Switchboard (SWB) datasets [108]. The main network is trained with single speaker segments based on the speaker labels available in either dataset. Train, heldout and test speaker sets are disjoint and include 2201, 275, and 275 speakers for BN and 519, 90, and 40 for SWB, respectively. Total durations of the subsets are given in Table 3.1.

In order to construct training data with change points, we first identify audio segments that have a change point within them. Speech segments that border the change point are limited to be no more than 1 s apart. This allows us to filter out examples with advertisement, music or large segments of silence in between speech regions. We then create utterances that contain the change point and span several words to the left and right of the change point, possibly ending up with incomplete sentences. The average duration of an individual utterance with a change point is 11.8 s/11.5 s/10.3 s for BN train/heldout/test data, and 8.29 s/8.24 s/7.67 s for SWB train/heldout/test data. Table 3.2 and Table 3.3 show the total duration of the training, heldout and test sets (in hr) and also report the average durations of the first and second speakers per utterance (in s) along with their standard deviations for the BN and SWB data, respectively. In order not to bias the system from detecting changes only towards one side of the midpoint, we have tried to balance the average duration per side as shown in the third and fourth columns of Tables 3.2 and 3.3. For BN, when sampling utterances with change points, we end up with a speaker overlap between train and test

Table 3.2: Amount of training, heldout and test data for auxiliary network training and average (± stdev) speaker 1 and speaker 2 durations per utterance for the BN dataset

	Duration (hr)	S1 dur/utt (s)	S2 dur/utt (s)
Train	40.9	5.8 (± 2.2)	6.0 (± 2.2)
Heldout	20.6	5.7 (± 2.4)	5.8 (± 2.4)
Test	4.4	5.0 (± 3.1)	5.3 (± 3.2)

Table 3.3: Amount of training, heldout and test data for auxiliary network training and average (± stdev) speaker 1 and speaker 2 durations per utterance for the SWB dataset

	Duration (hr)	S1 dur/utt (s)	S2 dur/utt (s)
Train	152.5	3.95 (± 2.7)	4.34 (± 2.8)
Heldout	6.1	3.93 (± 2.7)	4.31 (± 2.9)
Test	1.1	3.46 (± 2.4)	4.21 (± 3.1)

sets. However, we make sure that at least one of the speakers in each test utterance is from the original test set (one of the 275 speakers in the test data shown in Table 3.1). For SWB, we sampled the change points from the Hub5-2000 test set [109] which was pre-segmented automatically at speaker change points for ASR.

Figure 3.4 provides an example utterance from the BN dataset which is a part of the following dialog:

– A: I'm very disheartened to hear that it has been grounded.

– B: Do you feel safe when you fly?

Figure 3.5 shows the spectrogram of the waveform shown in Fig. 3.4. Especially in the spectrogram, it hard to discern the speaker change by a simple inspection. In our problem, we are not given the exact change point, yet, we are trying to adapt on-the-fly even if the first and last parts of the utterance are spoken by different people.

The input speech features that we use are 40-d log-mel features normalized by a global mean and variance normalization followed by utterance-level mean normalization. The main network consists of three LSTM layers followed by two FC layers and a softmax layer for the outputs. LSTMs are unidirectional and they have 128 units per layer for BN and 256 for SWB.

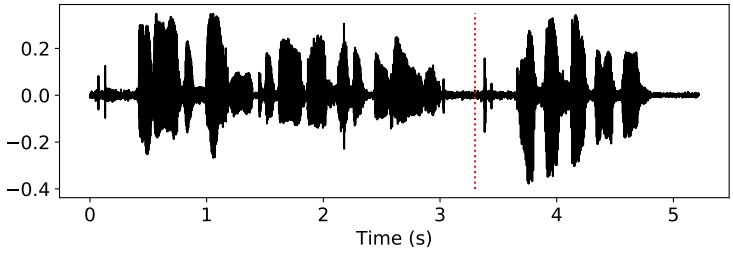


Figure 3.4: Waveform of an utterance with a speaker change point. The red line shows the change point location.

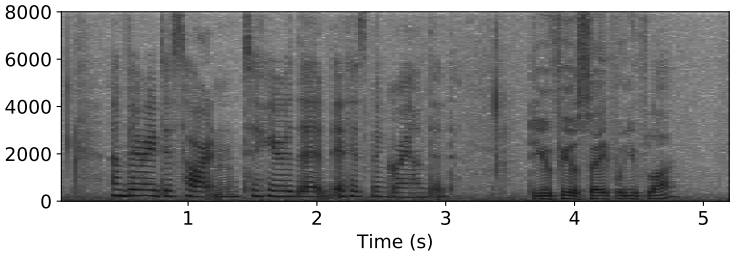


Figure 3.5: Spectrogram of the utterance shown in Fig. 3.4

The widths of the FC layers are 256 and 512, and the output layer size is 2000 which is also the number of senones modeled by the system. The main network is trained for 40 epochs with a decaying learning rate after the 10th epoch. The initial learning rate is set to 0.001. Main network parameters are optimized using the Adam optimizer [110] with the cross-entropy objective. PyTorch [111] is used for neural network training.

The auxiliary network consists of three FC layers with rectified linear unit nonlinearity except the last layer which has linear activations. The layer widths are 512, 256 and 128. With these settings, the sizes of W_w, W_b become 128×128 for BN, 256×128 for SWB, and the sizes of b_w, b_b are 128 or 256.

The main network is first trained on single speaker segments before we split the network to the lower M_L and upper M_U parts and attach the auxiliary network to the main network such that it takes input from M_L. Training of the auxiliary network is done on segments with one speaker change point in them. However, during training we do not explicitly use information about the change point location; instead, attention-based speaker vectors are computed during training (Eq. (3.5)) with $T_s = T_w = 100$ frames.

Training of the auxiliary network is done under two conditions: 1) the

main network is frozen and only the parameters of the auxiliary network are trained using the gradient flow from the main network output layer, 2) we fine-tune M_L along with the auxiliary network parameters using the gradient flow from the main network output layer.

We evaluate the system performance in terms of its speaker adaptation capabilities for ASR and also its accuracy in detecting change points at the output of the auxiliary network. In the following subsections, we present these two sets of results.

3.4.1 Speaker Adaptation for ASR

Table 3.4 shows the word error rate (WER) of the main and the augmented networks on both the BN and SWB test sets on test utterances with change point in them. The first row shows the performance when we do not apply any adaptation strategy to the main network which is trained only on single speaker segments. This unadapted main network achieves 14.3% test WER on BN and 17.1% on SWB if tested on single speaker segments (Table 3.1 test set). However, as shown in the table, when tested on segments with a change point (Table 3.2 and Table 3.3 test sets), the performance degrades to a WER of 23.2% for BN and 25.1% for SWB. After training the main network, we choose the adaptation layer, here $l = 1$, and split the main network into lower M_L and upper M_U parts. Additional experiments showed that $l = 1$ is the most effective layer which aligns with the observations made in [32]. As we will investigate a speaker dependent transformation of the main network activations, in the second row of Table 3.4, we show the performance when we add an additional nonlinear transformation layer to the main network. This layer is applied to the output of layer l and consists of an affine transform followed by tanh nonlinearity. This increases the number of parameters of the network and reduces the WER on the test data. In the third and fourth rows, we fine-tune either M_L alone or along with M_U on multi-speaker segments, without the additional nonlinear layer in the main network. This adaptation achieves about 1.7% absolute reduction in the WER for BN and 2.5% for SWB. We then compare the performance of the augmented system which includes the auxiliary network. If we freeze the main network and train only the auxiliary layers, we get 21.8% for BN and 22.1% for SWB which are

Table 3.4: WERs of the unadapted and adapted models on the test segments with a change point

Network	Adaptation	BN	SWB
Main	-	$23.2^{\ddagger}$	$25.1^{\ddagger}$
Main	Nonlinear transform layer	$22.4^{\ddagger}$	$22.6^{\dagger}$
Main	Fine-tune M_L	$21.9^{\ddagger}$	$22.6^{\dagger}$
Main	Fine-tune M_L + M_U	$21.5^{\ddagger}$	$22.6^{\dagger}$
Augmented	Auxiliary net	$21.8^{\ddagger}$	22.1
Augmented	Auxiliary net + fine-tune M_L	**20.9**	**21.6**

Table 3.5: WERs of the adapted models on the test segments according to the transformation type: mean removal (Eq. (3.6)), affine transform (Eq. (3.9)), nonlinear transform (Eq. (3.10))

Adaptation	Transform	BN	SWB
Auxiliary net	Mean removal	22.0	22.7
Auxiliary net	Affine	21.8	21.8
Auxiliary net	Nonlinear	21.8	22.1
Auxiliary net + fine-tune M_L	Mean removal	$21.7^{\ddagger}$	22.0
Auxiliary net + fine-tune M_L	Affine	$21.3^{\ddagger}$	21.8
Auxiliary net + fine-tune M_L	Nonlinear	**20.9**	**21.6**

comparable to fine-tuning M_L of the main network. If we fine-tune M_L along with the auxiliary network, then we further reduce the WER to 20.9% for BN and to 21.6% for SWB which are even lower than the case where we fine-tune the whole main network on the training data with change points. These results show that the proposed approach achieves adaptation significantly better than the unadapted network and it achieves 0.6% and 1.0% absolute reduction in WER as compared to fine-tuning the main network on data with speaker change for BN and SWB datasets, respectively. In this table and in the subsequent WER tables, we computed the statistical significance between the best system in a given column and the rest. We denote statistically worse systems with superscript † and ‡ at a p-value of (< 0.01) and (< 0.001), respectively. These computations are based on the Matched Pair Sentence Segment (MAPSSWE) test of the sc_stats software [112].

In Table 3.5, we investigate the type of transformation used on the LSTM output. As discussed in Section 3.3, we either have a segment-dependent mean removal (only bias adaptation), an affine transformation or a nonlinear

Table 3.6: WERs of the augmented model on the test segments based on the choice of the change point in the auxiliary network

Adaptation	Change point type	BN	SWB
Auxiliary net	True	21.8	21.9
Auxiliary net	Soft	21.8	22.1
Auxiliary net	Hard	22.1†	22.3
Auxiliary net + fine-tune M_L	True	**20.9**	22.2
Auxiliary net + fine-tune M_L	Soft	**20.9**	**21.6**
Auxiliary net + fine-tune M_L	Hard	21.2†	22.3†

transformation. As shown, WER improves when M_L is fine-tuned; either with or without fine-tuning, the affine transform and the nonlinear transform have lower WER than the mean removal (21.7% $\rightarrow$ 20.9% for BN with fine-tuning and 22.0%$\rightarrow$21.6% for SWB with fine-tuning). In the sequel, unless otherwise stated, we will use the nonlinear version of adaptation since it resulted in the lowest WER on both datasets.

Next, we compare the effect of the segmentation type on the augmented model in Table 3.6. In these experiments, we either used the ground truth speaker boundary to get the average activations from the auxiliary network, or we used the soft alignments as described in Section 3.3, or we hardened the soft alignments to get a change point. In the first and third cases where we have an explicit decision about the change point, say at frame $I * T_s$, we now have only two windows rather than N and hence we compute two averages, one for the first speaker s_1 and one for the second speaker s_2 in the utterance:

$$s_1 = \sum_{j=1}^{I} z_j \tag{3.11}$$

$$s_2 = \sum_{j=I+1}^{N} z_j. \tag{3.12}$$

Equations (3.7)-(3.10) are then applied to s_1 and s_2. In the soft alignment case, we allowed each segment to generate a separate speaker embedding using the attention mechanism of Eqs. (3.3)–(3.5). Decision hardening in the third case is achieved by taking the least similar consecutive segments.

Table 3.6 shows the WERs of the augmented systems depending on the

choice of change point type (true/soft/hard) for both datasets. Note that in a real test condition, true change points will not be available and hence this first case is not practical. As shown in Table 3.6, with soft alignments, we achieve WERs similar to those of the case where we use the ground truth boundaries for BN. The WER difference between the true and soft alignments is on the order of 0.01%, and hence it is not visible in the table for BN, whereas for SWB we see that soft alignments perform better than true changepoints on the SWB data. This shows an advantage of the proposed approach where we do not explicitly need the true change point during test time. Hardening of the soft alignments corresponds to finding the consecutive frame pairs that are the least similar with respect to the inner product. The precision of hard change points is not as good as the ground truth, which increases the WER as compared to the true change points up to 0.3% absolute (21.8→22.1 for auxiliary net on BN). When we compare soft versus hard decisions, we see that soft decisions outperform hard decisions significantly. We hypothesize that this is probably the result of a training and test mismatch, given that training is always done with soft change points. When we compare auxiliary network adaptation with and without fine-tuning of M_L, we see that fine-tuning uniformly lowers WER for soft change points, and is found to be statistically significant at the level of $p<0.01$, but does not always lower WER in the case of hard or true change points.

3.4.2 Speaker Change Detection

Given that the soft change detection performs well in terms of WER as shown in Table 3.6, we now evaluate the speaker change detection accuracy of the proposed system and compare it to three different speaker segmentation methods including the Siamese network based, i-vector based and x-vector based segmentation.

For the Siamese embedding and i-vector systems, experiments on the Broadcast News dataset used the best model architecture from [105], trained on 2 s-long segments. For SWB experiments, we trained a new Siamese model from scratch on the SWB dataset. Given the short duration of various segments, the i-vector based method tends to result in worse accuracy than the Siamese one. For both of these methods, we take our multi-speaker test seg-

ments and extract 2 s-long windows with a shift of 1 s. Then for each window we either feed them to the Siamese network or, for the i-vector case, concatenate the i-vectors of consecutive windows and feed to the same/different classifier. Based on the classifier decisions, we determine the change points.

For the BN x-vector system, we use the scripts from the Kaldi [113] x-vector example package to train an x-vector model on the BN single speaker segments. We then use the trained model to extract x-vectors and segment the test data using PLDA scoring and clustering scripts. Although during test time we give the true number of speakers within an audio file (which consists of several test segments), the x-vector system tends to oversegment the test data. On the other hand, for SWB data, we make use of a pretrained model available online [35, 114, 115]. Note that this model is trained not only on SWB data but also on NIST SRE data and on additional copies of the training data by adding noise and reverberation to the original datasets [35, 114, 115].

In this subsection, we will argue that our proposed system not only allows for ASR adaptation but also performs speaker change detection. For change detection, we neglect the M_U part of the network and only take the last hidden layer of the auxiliary network. Speaker change detection accuracy is measured as the fraction of speaker change points detected within 1 s of the true changepoint, averaged over the multi-speaker test segments. The speaker embedding layer of our joint adaptation network is used to compute two different types of speaker changepoints. In the "unconstrained" case, a speaker changepoint is detected every time the dissimilarity between two consecutive speaker embeddings, as computed by the same/different classifier network, exceeds a threshold; thus, any given multi-speaker test segment may have zero, one, or more than one detected changepoints. In the "constrained" case, exactly one speaker change point is detected per test segment, at the start time of the window with the largest speaker embedding dissimilarity.

Table 3.7 compares the accuracy of our proposed method with those of the Siamese embeddings [105], i-vector and x-vector [35] based speaker change detection for BN data. In this table, we report accuracy for both constrained and unconstrained conditions. In the unconstrained case, we do not constrain the number of segments within an utterance which may result in either no detection or oversegmentation. In the constrained case, we make sure that exactly one change point is detected. This constraint is imposed to the

Table 3.7: Speaker change detection accuracy (%) of x-vectors, i-vectors, Siamese network of [105] and the proposed systems on BN data

Segmentation System	Unconstrained	Constrained
x-vector	38.2	65.0
i-vector	43.0	63.8
Siamese net	45.5	64.2
Auxiliary net	51.4	68.2
Auxiliary net + fine-tune M_L	**51.5**	**71.3**

Table 3.8: Speaker change detection accuracy (%) of x-vectors, i-vectors, Siamese network of [105] and the proposed systems on SWB data

Segmentation System	Unconstrained	Constrained
x-vector (augmented training)	40.2	85.7
i-vector	25.9	69.7
Siamese net	25.6	72.6
Auxiliary net	46.6	70.5
Auxiliary net + fine-tune M_L	39.2	73.1

x-vector system by choosing the most dissimilar pair based on the PLDA scores. For i-vector and Siamese networks, the decision is made by selecting the different pair with the highest confidence score. For the proposed method, we choose the least similar pair of windows. Since the unconstrained case has many false positives, it results in lower accuracy than the constrained case. As shown in the table, in both cases, the proposed methods achieve higher accuracy. Although the alternative methods are trained explicitly to perform speaker change detection task or classification task, our proposed method, which did not use the explicit speaker boundaries during training time, performs better than the explicit models in both cases. Especially, in the unconstrained case, we get up to 34.8% (38.2 $\rightarrow$ 51.5) relatively higher accuracy as compared to the x-vector setup.

Table 3.8 shows the speaker change detection performance of systems on SWB data. The proposed method outperforms i-vectors and the Siamese network, but does not outperform x-vectors. As noted in the table, this x-vector embedding for SWB was trained using additional data from the NIST SRE datasets, and using data augmentation [35, 114, 115]; the ability of the x-vector system to leverage external datasets and augmented pre-training is an advantage not offered by the proposed system.

3.4.3 Discussion

In conventional ASR systems, audio files with multiple speakers are first segmented at speaker boundaries before decoding. This helps reduce errors, especially those from feature normalization mismatch when mean-variance statistics are estimated across speaker segments that have different acoustic characteristics. Decoding multi-speaker speech input without any pre-segmentation (`decode`([speakerA - speakerB]) where "-" denotes speaker change) versus concatenating the outputs of `decode`([speakerA]) and `decode` ([speakerB]) can hence result in different WERs. The latter case, where separate speaker based decodes are performed after segmentation, usually performs better with a lower WER. In the following discussion, we compare this two-step approach, i.e. segmentation followed by ASR, with the proposed method which can handle inputs with speaker change points. Given the inherent soft speaker attention mechanism integrated in the proposed method, we hypothesize that the auxiliary network system can effectively decode multi-speaker utterances without the need of an external segmenter.

In order to segment the test data with speaker change points, we use the x-vector based, i-vector based, Siamese network based decisions and hard decisions generated from the soft change points of the auxiliary network as discussed in Section 3.4.2 from the constrained setting. Once we get the single-speaker segments, we decode these with the unadapted main network and report results. We also use this change point information and perform a guided decoding (which was described in Eqs. (3.11) and (3.12)) of the best adapted model (Auxiliary network with fine-tuning of M_L). Finally, we compare these results with the soft alignment based decoding.

Table 3.9 shows the WERs on the BN test set. The first column denotes the type of segmenter, the second column is the set of WER results from the unadapted model and the last column is from the proposed model. For comparison, we include a final row, which shows the proposed joint soft change detection and adaptation result. This result is identical to the last row of Table 3.4. The main observation is that segmentation followed by unadapted ASR performs at least 0.9% worse in absolute terms than using the adaptive model. Differences in the change detection accuracy affect performance as seen in the WER differences while using the unadapted model. When decoding with the proposed model, however, we do not see significant differences

Table 3.9: WER with hard changepoints vs. soft changepoints: Speaker segmentation followed by speaker adaptation, BN test data. Last row is the soft-changepoint WER, from Table 3.4.

Segmentation system	Unadapted model	Proposed system
x-vector	23.2‡	21.2
i-vector	22.6	21.2
Siamese net	22.7†	21.1
Auxiliary net	22.4†	21.1
Auxiliary net + fine-tune M_L	22.1	21.2
Soft changepoints	-	**20.9**

across various segmentation systems, indicating that the network is robust to the precision of the change detector. Another observation is that the proposed system with soft detection achieves even lower WER than the two-step approach (the second or third column of the Table 3.9 versus 20.9%).

Table 3.10 shows the ASR performance based on the same five speaker segmentation methods for the SWB test set. Again, we see that adapted ASR has at least 1.6% better performance than the unadapted ASR which takes segmented speech signals generated by various diarization systems. Even if we had a good segmentation, for example with the augmented x-vector system, the unadapted ASR performs at 24.3% WER. On the other hand, when we use the proposed adaptation strategy, even without having a good segmenter, we achieve 22.3% WER, which corresponds to a 2% absolute WER gain. Moreover, if we use the proposed system with soft changepoints, we get 21.6% which brings 0.7% additional absolute reduction in WER. In terms of statistical significance, there is not a difference among i-vector, Siamese and auxiliary networks. However, at the level of p=0.01, we have statistically significant improvement between using hard change points versus soft decisions (22.3% versus 21.6%).

These results combined with the speaker change results show that with the proposed method, a single neural network can be used to transcribe speech, while at the same time, implicitly detecting change points. This is in contrast to conventional two-step approaches where we first explicitly segment the utterances and then use an ASR system for transcription. Additionally, the WER performance of the proposed method is also better than the traditional two-step approach, making the approach suitable also for online decoding

Table 3.10: WER with hard changepoints vs. soft changepoints: Speaker segmentation followed by speaker adaptation, SWB test data. Last row is soft-changepoint WER, from Table 3.4.

Segmentation system	Unadapted model	Proposed system
x-vector (augmented training)	24.3	22.7‡
i-vector	27.0‡	22.2
Siamese net	27.0‡	22.0
Auxiliary net	27.3‡	22.2
Auxiliary net + fine-tune M_L	27.2‡	22.3‡
Soft changepoints		**21.6**

applications like closed captioning of broadcast news where both ASR and speaker change detection are needed in a single pass.

CHAPTER 4

COUNTERFACTUALLY FAIR SPEAKER ADAPTATION

This chapter will introduce a counterfactual training method for speaker adaptation of E2E ASR systems. Although there are studies on counterfactual fairness in computer vision [93, 94] and natural language processing [95, 96], to our knowledge, this is the first proposal for speech processing.

We formulate the speaker adaptation problem as the following: Suppose that each speaker in the dataset has a counterfactual twin from the opposite gender (or a different protected attribute) and speaking exactly the same words. Then, irrespective of the gender (or a protected attribute), we would like to identify the words being spoken in the same way. This formulation fits to the counterfactual framework and can be combined with individualized versions of group fairness measures. We start from the proposal of [116], and modify it for the ASR task and propose the counterfactual equal opportunity and the counterfactual equal posterior in this chapter.

4.1 Introduction

As an important machine learning application, ASR is subject to fairness concerns. Various studies have shown concerns regarding the performance gap between male and female speakers [7] as well as black and white speakers [9]. Because of the power of modern speaker adaptation methods [13, 28, 117], the unfairness of ASR is usually cast as a problem of unfair training corpora, e.g., the study in [8] describes the under-performance of ASR for female speakers as a natural consequence of the under-representation of women in ASR training corpora. Counterfactual fairness provides an alternative approach: if two speakers speak the same sentence, with the same prosody and articulatory clarity, counterfactual fairness suggests that they should achieve similar ASR outcomes.

In this chapter, we frame the speaker adaptation problem from a counterfactual fairness point-of-view. We train the ASR so that it generates equivalent output label distributions for counterfactual speakers whose voices have been resampled with different sensitive attributes, but are otherwise identical in every respect that is not causally dependent on the sensitive attribute.

We can summarize the main contributions of this chapter as follows:

- Introduction of a speaker adaptation algorithm using an individualized counterfactual fairness criterion.
- Derivation of a counterfactually fair E2E ASR training method based on the sequence classification criterion CTC [40].
- Empirical comparison of three variants of our proposed counterfactually fair CTC, including an equal-odds variant based on [116], a method based on posterior matching, and a method based on matching the CTC loss.

4.2 Individual Equalized Counterfactual Odds

In [116], counterfactual fairness is combined with equalized odds in order to introduce individual equalized counterfactual odds:

$$P(\hat{Y}_{A\leftarrow a}|X=x, Y_{A\leftarrow a}=y, A=a) = P(\hat{Y}_{A\leftarrow a'}|X=x, Y_{A\leftarrow a'}=y, A=a), \quad (4.1)$$

which must be satisfied for all $x \in \mathcal{X}, y \in \mathcal{Y}$, and $a \in \mathcal{A}$.

The authors propose enforcing Eq. (4.1) using the counterfactual equal odds training criterion, which, for binary outcomes with tabular data, can be written as:

$$\begin{aligned}\mathcal{L}_{\text{CFEOdd}} =& J(\phi(x,a),y) + \lambda_{\text{CF}}\mathcal{L}_{\text{CF}} \\ &+ \lambda_{\text{CLM}} \sum_{a'} \mathbf{1}[a \neq a']\mathbf{1}[y = y_{A\leftarrow a'}](\Delta\sigma)^2, \end{aligned} \quad (4.2)$$

where J is binary cross-entropy, and $\phi(x,a)$ is the classifier output given observation x and sensitive attribute a. Weight parameters λ_{CF} and λ_{CLM} scale the contributions of the counterfactual loss ($\mathcal{L}_{\text{CF}}$) and the counterfactual logit matching loss, respectively. $(A \leftarrow a')$ denotes the *do* action, which

corresponds to generating the counterfactual, a and a' denote the true and counterfactual sensitive attribute, and $\mathbf{1}$ is the indicator function. Using σ^{-1} to denote the logit function, the terms $\mathcal{L}_{\text{CF}}$ and $\Delta\sigma$ are defined as:

$$\mathcal{L}_{\text{CF}} = \sum_{a'} \mathbf{1}[a \neq a'] J(\phi(x_{A \leftarrow a'}, a'), y_{A \leftarrow a'}), \tag{4.3}$$

$$\Delta\sigma = \sigma^{-1}(\phi(x_{A \leftarrow a'}, a')) - \sigma^{-1}(\phi(x, a)). \tag{4.4}$$

One point to note is that even though Eq. (4.1) uses conditioning on the true Y, the logit pairing term (Eq. (4.4)) does not explicitly depend on Y because when we consider the forward pass, the softmax output from the neural network will give us only the probability $P(\hat{Y}|X = x, A = a)$ rather than $P(\hat{Y}|X = x, A = a, Y = y)$.

Another observation is that the formulation above is for the binary classifier which means that equating the logit terms

$$\sigma^{-1}(\hat{Y} = 0|X, A) = \log \frac{P(\hat{Y} = 0|X, A)}{1 - P(\hat{Y} = 0|X, A)} = \log \frac{P(\hat{Y} = 0|X, A)}{P(\hat{Y} = 1|X, A)} \tag{4.5}$$

$$\sigma^{-1}(\hat{Y} = 1|X, A) = \log \frac{P(\hat{Y} = 1|X, A)}{P(\hat{Y} = 0|X, A)} = -\sigma^{-1}(\hat{Y} = 0|X, A) \tag{4.6}$$

between real and counterfactual inputs would mean matching the log-probability terms $\log P(\hat{Y}|X, A)$ since if we match either of Eq. (4.5) or Eq. (4.6), it will imply the other equality. Now, if we consider ASR or specifically the character recognition problem, we have a multi-class classifier ($K > 2$) and the logit terms become

$$\sigma^{-1}(\hat{Y} = k|X, A) = \log \frac{P(\hat{Y} = k|X, A)}{1 - P(\hat{Y} = k|X, A)} = \log \frac{P(\hat{Y} = k|X, A)}{P(\hat{Y} \neq k|X, A)}. \tag{4.7}$$

If we achieve the perfect match of the logits ($\forall k \in \{0, \ldots, K-1\}$), then this would again imply equality of log-probabilities resulting from the real and counterfactual inputs. However, during training the difference between real and counterfactual outputs will not be 0 for all k. Since the goal is to match probabilities and since it is also easier to compute the log-softmax as compared to the logits for the multi-class case, we will use the log-softmax outputs ($\log P(\hat{Y} = k|X, A)$) in the constraint rather than the logits in the experiments.

4.3 Individual Counterfactual Equal Opportunity

Using a similar reasoning as Eq. (4.1), we provide a relaxed version of equalized counterfactual odds which we call counterfactual equal opportunity:

$$P(\hat{Y}_{A\leftarrow a}=y|X=x, Y_{A\leftarrow a}=y, A=a) = P(\hat{Y}_{A\leftarrow a'}=y|X=x, Y_{A\leftarrow a'}=y, A=a). \tag{4.8}$$

This equation can be interpreted as only requiring similarity between the probabilities of correct outcomes (predicted outcome matches ground truth) given factual and counterfactual individuals. If, in Eq. (4.2), we replace the logit pairing term with the difference of the CTC losses between the factual individual, x, and the counterfactual individual, $x_{A\leftarrow a'}$, we arrive at the loss function for counterfactual equal opportunity for ASR:

$$\begin{aligned}\mathcal{L}_{\text{CFEOpp}} =& J(\phi(x,a),y) + \lambda_{\text{CF}}\mathcal{L}_{\text{CF}} \\ &+ \lambda_{\text{CCM}} \sum_{a'} \mathbf{1}[a \neq a']\mathbf{1}[y = y_{A\leftarrow a'}](\Delta\mathcal{L}_{\text{CTC}})^2 \qquad (4.9)\\ \Delta\mathcal{L}_{\text{CTC}} =& \mathcal{L}_{\text{CTC}}(\phi(x_{A\leftarrow a'}, a'), y) - \mathcal{L}_{\text{CTC}}(\phi(x,a),y), \qquad (4.10)\end{aligned}$$

where λ_{CCM} denotes the counterfactual CTC loss matching factor which is a hyper-parameter.

4.4 Counterfactual Posterior Matching

Let us revisit the proposal in Section 4.2. It tries closing the difference in $P(\hat{Y}|X,A)$ between the real and counterfactual outputs. Using conditional probabilities, we can write it as follows:

$$P(\hat{Y}|X{=}x, A{=}a) = \sum_{y\in\mathcal{Y}} P(\hat{Y}|X{=}x, A{=}a, Y{=}y)P(Y{=}y|X{=}x, A{=}a). \tag{4.11}$$

Here $\mathcal{Y}$ is the all possible transcriptions that we can get given X, A. For a signal with T frames this would mean all possible character sequences up to length T for the given set of K characters. This space is very large and it is computationally hard to compute all possible posteriors. One simplification is to assume that $P(Y|X{=}x, A{=}a) = \mathbf{1}[Y{=}\mathbf{c}|X{=}x, A{=}a]$ where $\mathbf{c}$ is the

ground truth character sequence. With this assumption, we will reach to the proposal of this section:

$$P(\hat{Y}|X{=}x, A{=}a) = \sum_{y \in \mathcal{Y}} P(\hat{Y}|X{=}x, A{=}a, Y{=}y)\mathbf{1}[Y{=}\mathbf{c}|X{=}x, A{=}a] \quad (4.12)$$

$$= P(\hat{Y}|X{=}x, A{=}a, Y{=}\mathbf{c}). \quad (4.13)$$

In other words, as in Eq. (4.1), the goal is to match the posterior probability of $\hat{Y}$ after observing the target outcome Y. With the above assumption, this corresponds to the probability of characters at the softmax layer after observing the true ground truth sequence. In the case of CTC, this would be the character posteriors $P(\hat{Y}_t = k|X = x, A = a, Y = y)$. Then, the objective function becomes

$$\begin{aligned}\mathcal{L}_{\text{CFPM}} =& J(\phi(x, a), y) + \lambda_{\text{CF}}\mathcal{L}_{\text{CF}} \\ &+ \lambda_{\text{CPM}} \sum_{a'} \mathbf{1}[a \neq a']\mathbf{1}[y = y_{A \leftarrow a'}](\Delta\gamma)^2 \end{aligned} \quad (4.14)$$

where

$$\Delta\gamma = \gamma(\phi(x_{A \leftarrow a'}, a'), y) - \gamma(\phi(x, a), y) \quad (4.15)$$

$$\gamma(\hat{y}, y) = P(\hat{Y} = \hat{y}|X = x, Y = y, A = a). \quad (4.16)$$

When the main objective function J is the CTC loss, then the posterior probability γ can be obtained using the forward and backward variables of CTC loss computation. Let $\alpha_t(s)$ and $\beta_t(s)$ denote the CTC path at time t passing through the index s of the blank symbol augmented ground truth label sequence l'. The posterior of passing through s at time t then becomes

$$\gamma_t(s) = \frac{\alpha_t(s)\beta_t(s)}{\sum_{s'} \alpha_t(s')\beta_t(s')}. \quad (4.17)$$

In order to get the character (k) posteriors at each time index t, we need to sum over s indices such that $l'(s) = k$:

$$\gamma(\hat{y}_t = k, y_t = k) = \sum_{s:l'(s)=k} \gamma_t(s). \quad (4.18)$$

Table 4.1: Duration of male and female speech in hours depending on the data split

Subset	Male	Female	Total
Train	9.8	11.2	21
Dev	2.5	2.2	4.7
Test	2.8	1.9	4.7

In the experiments, we directly try matching the $\gamma_t(s)$'s (or $\log \gamma_t(s)$) as they will lead to equality of $\gamma(\hat{y}_t = k, y_t = k)$. We will test the simplification mentioned above by comparing Eq. (4.14) to Eqs. (4.2) and (4.9). We will see that the simplification hurts the final performance but it still works better than the equal opportunity relaxation of Section 4.3.

In order to train an E2E ASR model with the above objectives, we need the counterfactual counterparts of each utterance in our dataset. This is a challenging subproblem because a) counterfactuals do not exist in the real world, b) especially in free-form speech such as interviews (as opposed to read speech), it is hard to obtain parallel datasets in which two different people utter the same words in a similar manner. Hence, we need to generate utterances as if the speaker were from the opposite gender while keeping the spoken content the same. In our experiments, we use the TimeGAN model proposed for generation of sequential data [118]. As this model explicitly uses the global attribute, it allows us to change the speaker gender label which is our sensitive global attribute per data point. Details of this model will be provided in the next section.

4.5 Experiments

We performed our experiments on the Corpus of Regional African American Language (CORAAL) [119]. The dataset is split into train, development, and test sets based on the speakers. All speakers in the datasets are alphabetically sorted and the utterances belonging to the first 64 male and 64 female speakers are used for training. From the remaining set, 8 male and 8 female speakers are used in development set and the remaining 14 male and 6 female speakers' utterances are used for test purposes. Table 4.1 summarizes the total duration per gender in terms of hours.

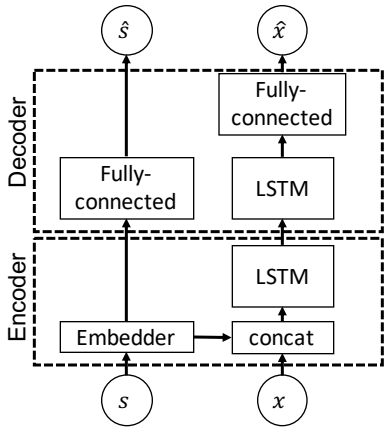


Figure 4.1: The auto-encoder model used to generate counterfactual examples

The baseline system is a DeepSpeech2 model [120] trained on the CORAAL dataset with CTC loss. Input features are log magnitude spectrograms extracted from 20 ms windows with 10 ms skip, and a Hamming window is used for shaping the time-domain data. Our network outputs are English alphabet characters along with blank, apostrophe and the end-of-sentence token. The baseline DeepSpeech2 model has two convolutional layers, each with batch normalization and tanh activation. The convolution kernel sizes are 41×11 and 21×11 respectively. These layers are followed by 5 batch-normalized bidirectional LSTM layers with 768 cells, whose output is fed into an FC layer. The baseline model is trained for 30 epochs with Adam optimization, batch-size 16 and learning rate of 0.001. All models are implemented using PyTorch [111] and each one ran on a single Nvidia Tesla V100 GPU.

In order to generate the counterfactual inputs for training, we used the auto-encoder part of the TimeGAN model [118]. As shown in Fig. 4.1, this model takes the input audio features (x) and a global attribute (s) of the input, encodes them in to hidden vectors and then tries recovering both the input features and the attribute separately. In our implementation, the encoder takes one-hot representation of speaker gender, and uses an embedding layer. These 256 dimensional embedded speaker vectors are then appended to the acoustic features which in turn are passed through a 2-layer LSTM network with 256 cells per layer. The decoder takes the output of the encoder and generates two types of outputs: A 2-layer FC network with ReLU

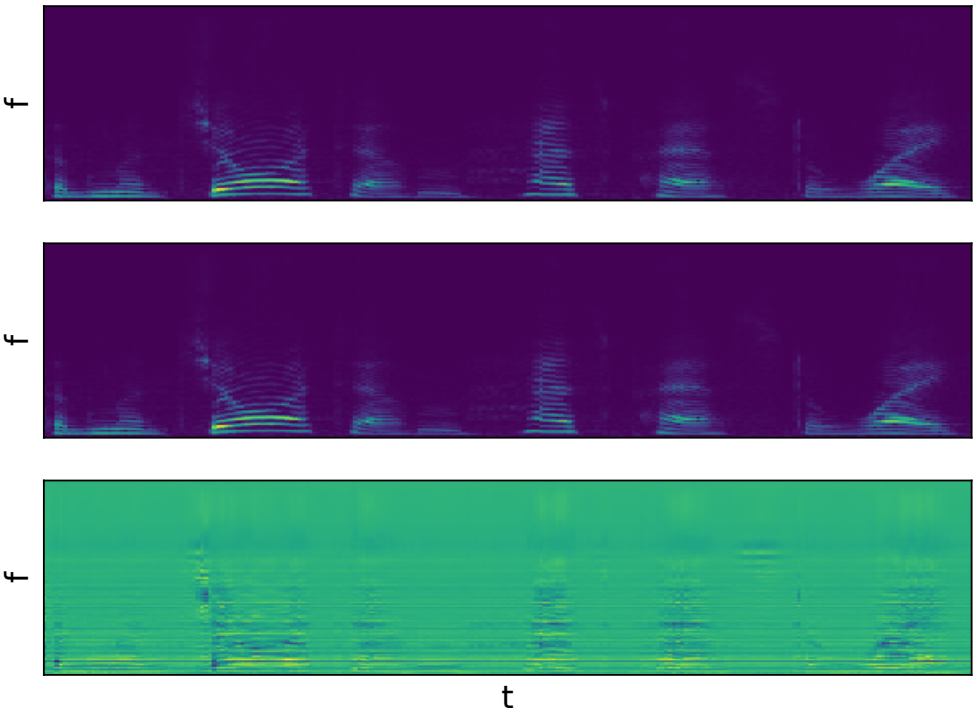


Figure 4.2: Reconstructed spectra (t: time, f: frequency) for a sample development set utterance. Top row: reconstruction based on the true gender (female) attribute, Middle row: reconstruction when we change the gender to male, Bottom row: difference between first two rows.

activations maps speaker embeddings back to the one-hot representation ($\hat{s}$), while in the second output layer ($\hat{x}$), we map hidden vectors into acoustic feature vectors by processing them with a 2-layer LSTM with 256 cells per layer followed by a FC layer. This network is trained using a multi-task loss where the components are the L2-distance between reconstructed features and the ground truth features and the L2-distance between estimated and the ground truth speaker attribute one-hot vectors. This network is trained using batch-size 16, learning rate 0.001, with Adam optimization for 50 epochs on the CORAAL training set. Once this model is learned, counterfactual examples are generated by switching the gender variable in the one-hot speaker attribute vector and providing the original speech features as input.

The proposed ASR models are then trained on original features and the

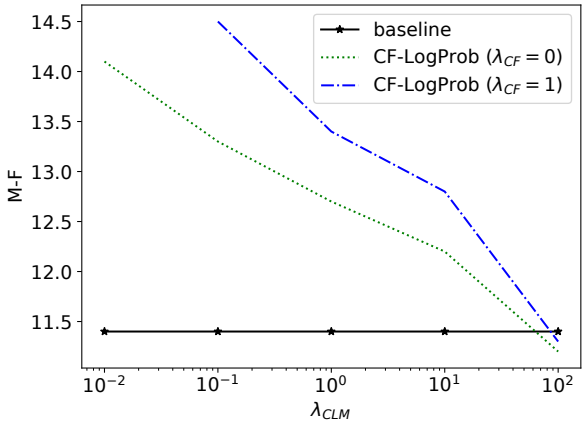


Figure 4.3: CER gap for genders from the unadapted model and log-probability matching approaches

counterfactual data. This model is a DeepSpeech2 model similar to the baseline but trained from scratch with the proposed objectives instead of just CTC.

We compared the performances of the ASR systems based on the overall character error rate (CER) on the test data, the CER difference between male and female speakers and also the standard deviation of CER across all test speakers. We tested the significance of the CER difference between models using NIST's SCLITE toolkit, MAPSSWE method [112] and reported models with significant change at p-value of 0.001 when applicable.

4.5.1 Results

In this section, we will give an example input pair for our counterfactual training algorithm, and then show the results from our ASR experiments.

In Fig. 4.2, we show example outputs from the model described in Fig. 4.1. The top spectrogram belongs to reconstructed version of the original utterance from a female speaker. The middle one shows the spectrogram when we abduct bottleneck autoencoder features from the female speaker, change the (s) variable to male, and then reconstruct. Since it is hard to see the differences between these two spectra, we also include the difference between them in the bottom row. It can be observed that differences between the

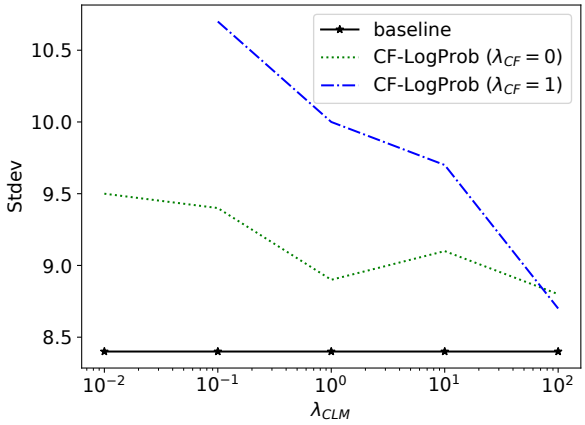


Figure 4.4: Standard deviation of the CER over all speakers from the unadapted model and log-probability matching approaches

spectrograms are near zero at most frequencies, but show negative and positive deviations from zero in closely spaced narrow frequency bands where the voice conversion has shifted energy to a lower frequency in order to model the shift from female to male.

Initially, we performed two types of ASR experiments. In the first set of experiments, our aim is to determine if the middle term in Eqs. (4.2), i.e., the CTC loss due to the counterfactual input, is crucial. The counterfactual log-probability matching model (Eq. (4.2)) is trained under two conditions, $\lambda_{\text{CF}} \in \{0, 1\}$, each while sweeping the log-probability matching weight (λ_{CLM}) from 0.01 to 100. Figures 4.3 and 4.4 show the CER difference between males and females, and the standard deviation of CER across test speakers, respectively. As we observe from these figures, the log-probability matching approach, including the loss term due to the counterfactual input (denoted as $\lambda_{\text{CF}} = 1$ in the legend), has a larger gap between gender groups and also a larger standard deviation. Therefore, in the subsequent experiments, we set $\lambda_{\text{CF}} = 0$. Interestingly, irrespective of λ_{CF}, for most values of λ_{CLM} that we have tested, the gap and the standard deviation were higher as compared to the baseline model which was only trained with CTC loss on the original input features. Reduced male-female gap was only observed when we increased λ_{CLM} over 100.

In the second set of experiments, our goal is to compare the unadapted

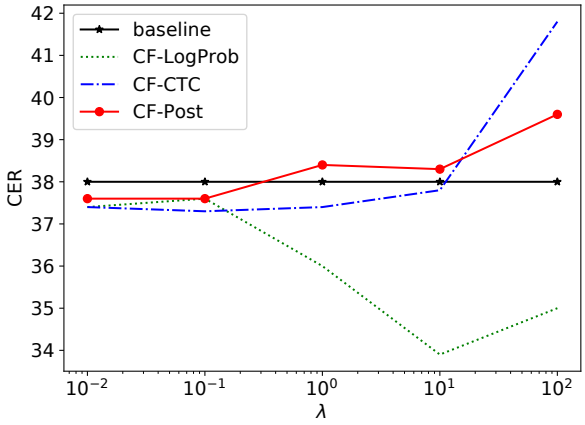


Figure 4.5: CER for the unadapted model, counterfactual log-probability matching, CTC matching and posterior matching approaches

baseline, counterfactual log-probability matching (CF-LogProb), counterfactual CTC loss matching (CF-CTC) and the proposed log character posterior matching (CF-Post) models. As mentioned above, here we set $\lambda_{\text{CF}} = 0$ and only sweep the λ corresponding to the last term in Eqs. (4.2), (4.9) and (4.14) (respectively λ_{CLM}, λ_{CCM}, or λ_{CPM}). Figure 4.5 compares the average CER of the four models as a function of λ. Figure 4.6 shows the CER gap between male and female, and Fig. 4.7 shows the inter-speaker standard deviation of CER.

In terms of the overall CER (Fig. 4.5), only the log-probability matching approach achieves significantly lower CER than the baseline for most values of λ. The log posterior matching performs similarly to the baseline for small values of λ, but when λ reaches 100, it performs significantly worse than the baseline. On the other hand, the CTC loss matching approach results in a large increase in the CER as we increase the weight of the counterfactual fairness term λ.

In Fig. 4.6, we compare the CER gap between males and females. Although the CTC matching approach has a lower gap (fairer) than the other two approaches, it has much higher overall CER (lower accuracy) as we show in Fig. 4.5. The LogProb approach performs similarly to the baseline at $\lambda = 100$ whereas the log-posterior matching has slightly higher gap than the baseline.

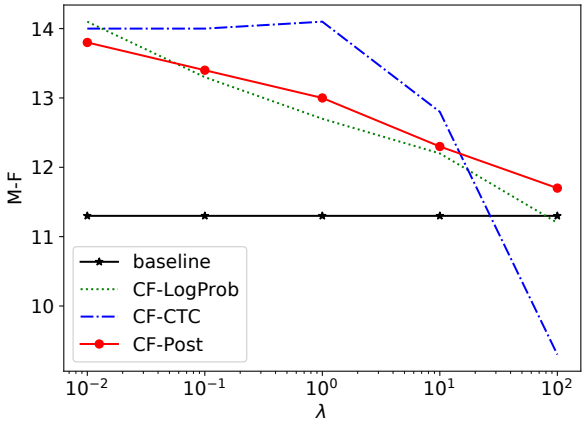


Figure 4.6: CER gap for genders from the unadapted model, counterfactual log-probability matching, CTC matching and posterior matching approaches

In Fig. 4.7, we compare the standard deviation of CERs from different models. As expected, the curves usually have downward slopes because as we increase the weight of fairness λ, we should achieve fairer outcomes, i.e. lower inter-speaker standard deviation. When $\lambda < 10$ all three approaches have higher standard deviation as compared to the baseline and the posterior matching approach has the smallest deviation. When $\lambda = 100$, CTC and posterior matching approaches perform better than the baseline but they come at the expense of having higher CER as shown in Fig. 4.5. Although the log-probability matching method has lower male-female gap in Fig. 4.6 at $\lambda = 100$, we see that the standard deviation of CER is not lower than the baseline.

Although the log-posterior approach has a higher standard deviation, it has smaller CER increase as compared to the CTC matching approach. This might be related to the fact that equal opportunity (CTC matching) is weaker than the equal odds (log-posterior matching) criterion. If we compare log-posterior vs. log-probability matching, even though the latter has higher standard deviation at $\lambda = 100$, the CER is still lower than the baseline so it might be argued that the log-probability matching approach is the most effective one in practice. Hence, the remaining experiments will be based on the log-probability matching approach.

As the inter-speaker standard deviations of the log-probability matching

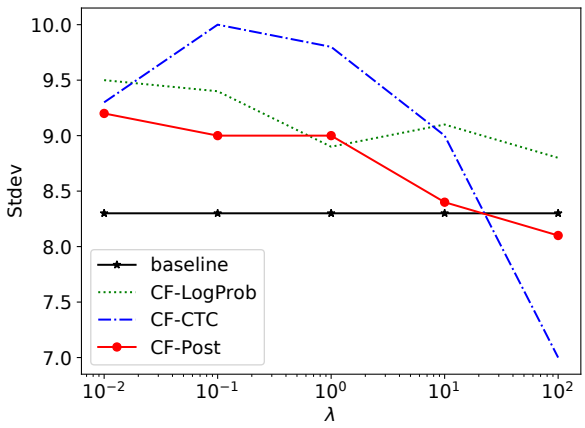


Figure 4.7: Standard deviation of the CER over all speakers from the unadapted model, counterfactual log-probability matching, CTC matching and posterior matching approaches

experiments are higher with a decreasing trend, we also investigated larger values of λ_{CLM}. As shown in Figs. 4.8-4.10, for $\lambda = 200$ and 300, we still operate at a CER value lower than the baseline while improving the fairness in terms of the male-female CER gap and the standard deviation. Especially, $\lambda = 300$ provides a good operating point with the largest improvement in fairness while maintaining low CER.

In the experiments described above, the protected attribute was always gender. However, CORAAL dataset comes with the speaker metadata including their age and education groups which can also be considered as protected attributes. In the sequel, we will investigate the cases where the protected attribute is age or education group rather than gender. In these experiments, we still use the auto-encoder model described in Fig. 4.1 except that the number of possible attributes changes depending on the experiment. For example, in the age group experiments, we have 10 classes as there are 5 age groups from two genders. Having 10 classes instead of 5 allows us to keep the gender attribute the same while generating the counterfactual data from a different age group.

As we can see from Table 4.2, when $\lambda_{\text{CLM}} \in \{200, 300\}$, we operate at an equal or lower CER level than the baseline while reducing the inter-speaker standard deviation of the CER. Although we focus on age group as the protected group, we are able to reduce the gender gap in these experiments.

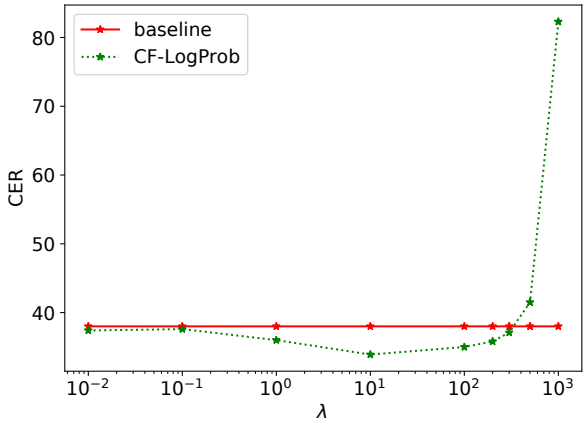


Figure 4.8: CER for the unadapted model, and counterfactual log-probability matching for various values of λ

Table 4.2: CER statistics from the counterfactual log-probability matching experiments where age is the protected attribute. CORAAL test data only contains speakers from age groups 2, 3 and 4.

Model	CER	Stdev	M	F	M-F	Age2	Age3	Age4
Baseline	38.0	8.3	44	32.6	11.4	39.5	42.7	38.9
CF-Age-$\lambda = 200$	36.2	8.2	41.6	31.2	10.4	37.1	40.9	36.7
CF-Age-$\lambda = 300$	38.0	7.8	43.2	33.3	9.9	38.9	42.7	38.3
CF-Age-$\lambda = 500$	45.5	6.8	49.9	41.6	8.3	46.5	49.9	45.2
CF-Age-$\lambda = 1000$	86.7	1.4	86.8	87	-0.2	87.2	87.7	85.7

When we look at the average CER per age group in the test set, we also see a decrease for $\lambda_{\text{CLM}} \in \{200, 300\}$. As the regularization gets stronger, i.e., λ gets larger, we further reduce the standard deviation but we observe higher CERs.

Next, we will investigate the case where the protected attribute is the education level of the speaker. According to the results shown in Table 4.3, when we have $\lambda_{\text{CLM}} = 200$, we operate at a lower CER than the baseline while having lower inter-speaker standard deviation. For comparison, we also include the male/female and age group statistics in each case. As we can see, when $\lambda_{\text{CLM}} = 200$, we are able to reduce the gender gap, as well as the CERs per age group. Since there are many education categories, those statistics are not provided in the table. However, if we look at the data,

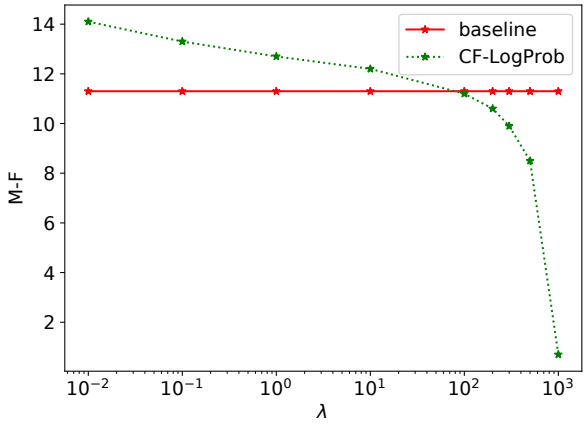


Figure 4.9: CER gap for genders from the unadapted model, and counterfactual log-probability matching for various values of λ

we also observe some decrease in CER for each education category. We will provide further discussion in the next section.

Table 4.3: CER statistics from the counterfactual log-probability matching experiments on CORAAL where education category is the protected attribute

Model	CER	Stdev	M	F	M-F	Age2	Age3	Age4
Baseline	38.0	8.3	44.0	32.6	11.4	39.5	42.7	38.9
CF-Edu-$\lambda = 100$	34.7	8.3	40.2	29.8	10.4	35.3	39.7	35.5
CF-Edu-$\lambda = 200$	37.5	7.6	42.4	33.3	9.1	38.0	42.4	37.8
CF-Edu-$\lambda = 300$	53.6	5.3	56.7	50.9	5.8	54.2	57.2	52.9

The experiments described above are on the CORAAL dataset. In the next experiment, we will experiment on a standard American English dataset, namely, 100 hr subset of LibriSpeech [121]. This dataset contains only gender information of the speaker, hence we will only test the performance when the protected attribute is gender. The experimental procedure is similar to the CORAAL dataset except that we train the auto-encoder on LibriSpeech train dataset. Table 4.4 shows the CER performance on LibriSpeech. Since the LibriSpeech is a larger dataset with read speech, in general, we operate at a lower CER level as compared to CORAAL. Since earlier, we observe that in counterfactual gender experiments, the optimal λ is 300, for LibriSpeech

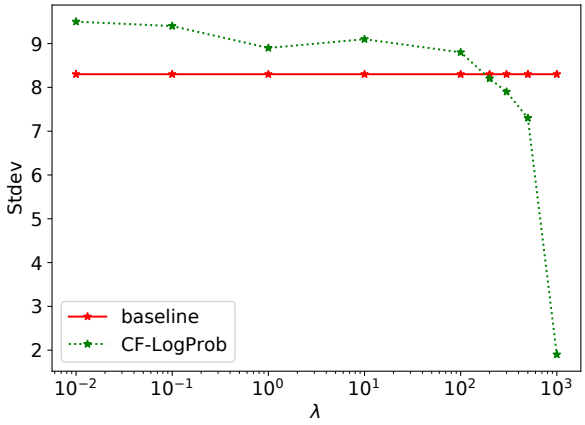


Figure 4.10: Standard deviation of the CER over all speakers from the unadapted model, and counterfactual log-probability matching for various values of λ

we also try values 300 and 500. When $\lambda = 300$, we do observe reduction in both the overall CER as well as the inter-speaker standard deviation. Furthermore, we reduce the gender gap from 1.7% to 1.4%.

Table 4.4: CER statistics from the counterfactual log-probability matching experiments on LibriSpeech when the protected attribute is gender

Model	CER	Stdev	M	F	F-M
Baseline	9.8	2.9	8.9	10.6	1.7
CF-Gender-$\lambda = 300$	9.6	2.6	8.8	10.2	1.4
CF-Gender-$\lambda = 500$	10.3	2.8	9.5	10.9	1.4

4.5.2 Discussion

Systems described above are trained with an individual fairness objective, but results are reported using group disparity (overall male CER vs. female CER). This is mainly because counterfactuals do not really exist. In order to compensate for that, we included the standard deviation across all speakers as a proxy for the individual differences. A full discussion of whether there is a trade-off between individual and group fairness is out of scope of this study and we refer to [89] for a relevant discussion. One way to investigate the im-

provement in individual differences is to look at the CER differences between a speaker and their counterfactual realizations. Next, we will visualize these results on CORAAL dataset.

In Figs. 4.11-4.13, we show the total CER difference between real and counterfactual categories where the categories are gender, age, and education, respectively. In each case, the left subfigure shows the absolute CER gap from the baseline system and the right figure shows that of the log-probability matching system. Colors in the figures are shaded such that same colors in left and right subfigures correspond to the same level of difference. In all three figures, we see that the model obtained from counterfactual training has lower CER differences between categories. The differences are an order of magnitude lower than that of the baseline.

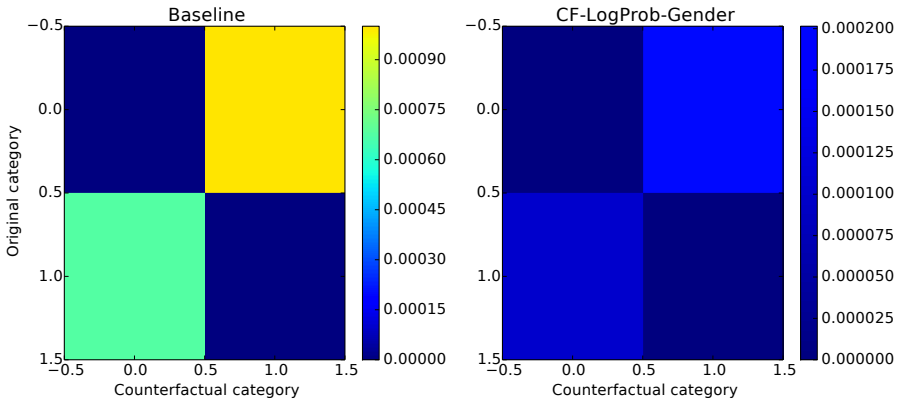


Figure 4.11: Average individual CER differences between real and counterfactual genders on CORAAL

Especially in the CTC loss matching experiments, we saw a trade-off between accuracy and fairness. Based on the discussion in [90] which states that "there exist ideal distributions where fairness and accuracy can be in accord," we may speculate that our train-test split does not follow the ideal distribution. If we look at Table 4.1, we see that the training and test sets are highly mismatched for two reasons: 1) test speakers are completely unknown during training, 2) the ratio of total duration of male and female speech differs in train and test subsets, in that training data have more female speech whereas the test data have more male speech.

It is interesting to note that although in standard American English datasets such as LibriSpeech (100 hr subset) [121], we observe that males have lower

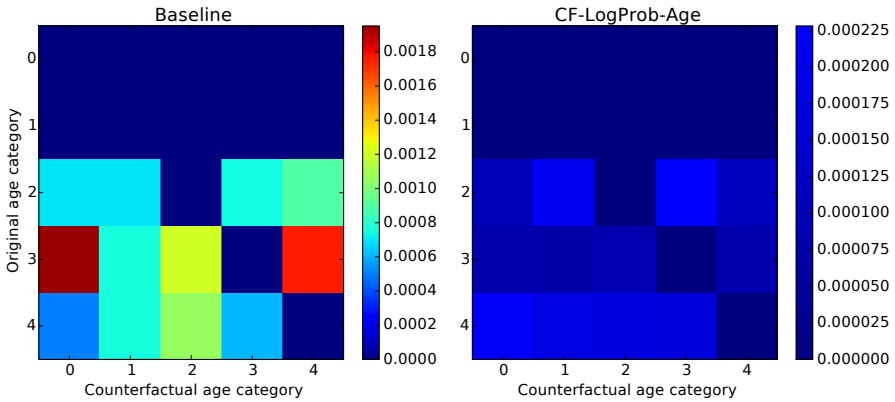


Figure 4.12: Average individual CER differences between real and counterfactual age groups on CORAAL

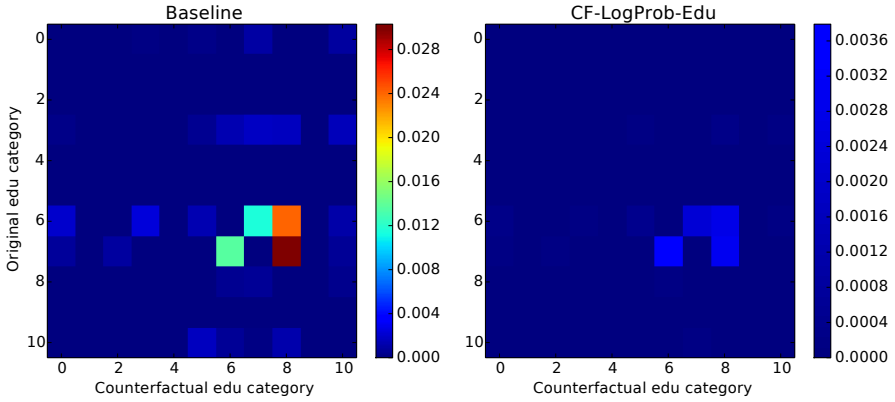


Figure 4.13: Average individual CER differences between real and counterfactual education groups on CORAAL

CER than females (8.9% vs. 10.6%, in our own experiments), in the African American dataset (20 hr) that we use, we observe the opposite: male speakers have 44% CER, while female speakers have 32.6%. One explanation for females having overall lower CER is that there is more female speech in the training set. Another explanation can be based on social linguistics; there are studies showing that female speakers will use a higher proportion of more standard forms than male speakers to avoid the stigma of a non-standard form, while male speakers may prefer to preserve their identity [122]. This may lead to heavier use of standard American by female speakers in CORAAL, as compared to a wider variety of African American

vernacular styles used by males. We provide example utterances from male and female speakers from both datasets below:

- CORAAL-male: Cause you know, we didn't never have to tan so we didn't never even sit out in the sun.
- CORAAL-female: But it's just not a safe space to have open and honest race discussions so I don't even go there.

– LibriSpeech-male: The music came nearer and he recalled the words of Shelley's fragment upon the Moon wandering companionless pale for weariness

– LibriSpeech-female: Then she gave Rosalie back her magic ring thanking the kind witch for all she had done for them.

In the CORAAL male utterance example, we see double negation, casual forms, etc. This observation is not specific to this particular example; we observe very frequent use of 'you know's or other casual forms among males. Another observation is that, in our test split, the average education level for females is higher than that for male speakers. As the education levels get higher, we usually see lower error rates. This imbalance could also explain the reason behind the lower error rates for females in CORAAL. The male-female gap for LibriSpeech is less obvious (only around 1.5%). This is partially due to having a balanced amount of speech from each group. One possible explanation for lower performance for females may be attributed to the observation that female speech is more variable within this group. For example, if we look at gender specific standard deviations of CERs, we see that they are 2.4 and 3.2 for males and females, respectively. This may support our hypothesis that female speech is more variable at least in this particular setting. The example utterances above also explain the performance level difference between CORAAL and LibriSpeech; the former is based on conversational speech from interviews whereas the latter is read speech obtained from book readings.

In summary, in this chapter, we investigated the ASR speaker adaptation problem from an individualized counterfactual fairness point-of-view. We propose that for any given individual, if this person were from a different protected group such as the opposite gender, but spoke the same words with

similar rhythm and intonation, fair ASR should estimate the same characters as its output. We formulated this as an additional loss term that is added to the CTC loss due to the original input. We compared three approaches: matching the log softmax output from the ASR model, matching the CTC loss and matching the log posterior of characters given the ground truth sequence. We argue that these last two correspond to the individualized counterfactual equal opportunity and equal odds, respectively. In the experiments on CORAAL, we showed that there is generally a trade-off between the CER and fairness of the system. Especially, in the case of CTC loss matching, the CER increased significantly while achieving fairness. On the other hand, in the log-probability matching experiments, for certain values of λ, we were able to operate at a lower CER while reducing the standard deviation (unfairness). We verified the effectiveness of the proposed adaptation approach for different protected attributes (gender, age and education level) on an African American dataset as well as on a standard American English dataset.

CHAPTER 5

END-TO-END SPOKEN LANGUAGE UNDERSTANDING

In this chapter, the focus is on E2E SLU. We will introduce a method that takes advantage of non-parallel text data to learn better speech-to-concept models. We will demonstrate the performance on the speech-to-dialog act and speech-to-intent classification. This chapter is mainly based on our previous work [123].

5.1 Introduction

Speech understanding is a major component of human-machine interactions and its quality affects the user experience. Conventional speech understanding systems rely on a two-step approach where the speech signals are converted into text using an ASR system and then an NLP system is applied to understand intents, to fill the slots or to detect named entities [62, 63]. However, this two-step approach suffers from error-propagation due to imperfect ASR systems and also from non-optimality as ASR and NLP systems are trained separately with different objectives. Moreover, for many of the world languages, there is insufficient data to train reliable ASR systems [124, 125]. Therefore, there is an interest in approaches which can directly use speech input to achieve the understanding task without using intermediate ASR transcripts [2, 68, 69, 70].

Given that the variability in speech signals is larger than that of the text inputs, and also the fact that recent text-based embeddings such as BERT [66] achieve state-of-the-art performance in NLP tasks, the performance of text based SLU systems is usually better than that of corresponding speech based systems. To improve the performance of speech-only based systems it would therefore be useful to utilize the complimentary information present in text based representations. For many training approaches,

although parallel speech and text data would be required to integrate such information, with multiview based techniques we can train systems with non-parallel data. Systems trained in this fashion also have an advantage of being able to use any one of the two modalities at test time.

In this chapter, we focus on two goals for learning SLU systems with non-parallel data using speech-only dialog act as an example task. First, we propose a multimodal (speech and text) approach for dialog act recognition based on a multiview training approach. In many practical scenarios, we have large amounts external text data but limited amount of speech data with the corresponding text for dialog act recognition. Therefore, our goal is to show how we can improve speech-only performance by incorporating text information during training, especially in the non-parallel text case in the multiview approach. The main idea of the proposed multiview system is that we try to tie the speech and text encodings using a shared classifier. Second, if we are given an ASR system during training time, we try to identify the best way of utilizing information in the ASR model to train a speech-based dialog act recognition system.

5.2 Multiview Training

As observed in earlier studies, achieving good performance in a speech-only E2E SLU system is difficult especially with limited amounts of data. It has also been shown that multimodal approaches usually improve results as compared to unimodal systems [126, 23]. When labeled text data is available for a task, we therefore hypothesize that it will be useful to improve the speech-based system.

One direct way of utilizing two modalities is to append the features in the system either at the input or at an intermediate level. However, training such a system requires parallel data corresponding to the same sample at both training and test time, whereas—especially during test time—we lack access to text data for speech-based dialog act recognition.

To handle the non-parallel data case, we propose a multiview learning technique which consists of two unimodal branches which are coupled. The unimodal systems take either text or speech as input and produce dialog act labels. They consist of an encoder and a classifier. In this work, we used

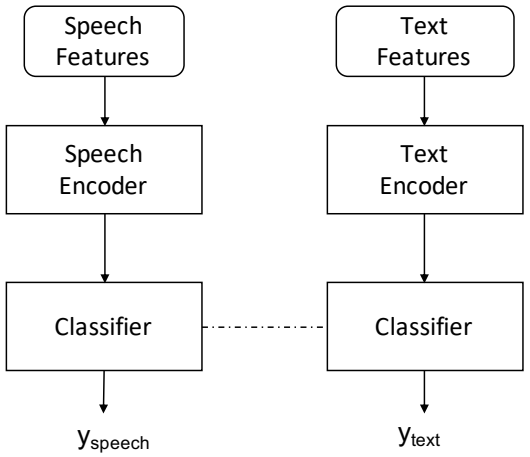


Figure 5.1: Multiview approach

BERT [66] embeddings as text features and MFCCs or ASR-derived acoustic embeddings as speech features to the unimodal systems.

Our proposed model shown in Fig. 5.1, processes speech and text information separately using two branches. We try to force the learned embeddings to be similar by using a shared classifier on both branches. The system can be thought of as an inverted Siamese network because of the shared classification parts in the two branches. This structure allows us to partially train the model using one modality without parallel speech and text data. This model is also practical as it allows to use speech data during test time.

Training of the multiview model is summarized in Algorithm 1. We start with training the encoder and classifier with the rich-resource (text) modality. Then we freeze the classifier and train the encoder on the other modality (speech) and in the final step we fine-tune both branches using parallel data while still sharing the classifier between the two branches. If there is no parallel-date available, we skip the fine-tuning step. At the end, we report the speech branch accuracy.

Algorithm 1 Training steps of the multiview system

Input: Labeled text-only, speech-only and parallel data

Output: Dialog act labels per utterance and overall accuracy

Train the text branch using text-only data
Freeze the classifier
Train the speech encoder with fixed classifier on speech-only data
if parallel data exists, **then**
    Fine-tune the encoders and the classifier on parallel data
end if
Test the speech branch alone
return Speech branch accuracy

5.3 Experiments

Experiments are performed on the Switchboard Dialog Act Corpus (SWDA) [127, 128]. The labels in the dataset are originally associated with text rather than speech. To use both speech and text modalities, we first create a matching speech corpus by finding the corresponding speech segments from the original Switchboard dataset based on forced alignments. Although we have parallel data in many practical settings, we only have non-parallel speech and text. We simulated this non-parallel setting by splitting the training data into text-only, speech-only and parallel portions where the amounts of total training, heldout and test sets are determined based on the division of [61].

In the first set of experiments, we used MFCCs with delta and double delta features as speech input. For text input, we extracted BERT embeddings [66] from a pretrained model on the true transcripts.

In multiview systems, the speech encoder consists of 3 BLSTM layers each of size 128 followed by 2 fully-connected layers of size 64. The text encoder consists of 2 BLSTM layers each with 128 units followed by a single fully-connected layer. In both branches transition from the BLSTM layer to the fully-connected layers is achieved by averaging over time. The classifier has 3 fully-connected layers with rectified linear unit nonlinearity.

Figure 5.2 compares the classification accuracies of four speech-only systems depending on the amount of non-parallel (NP) speech data used in training. The baseline is the case where we train the speech branch on low

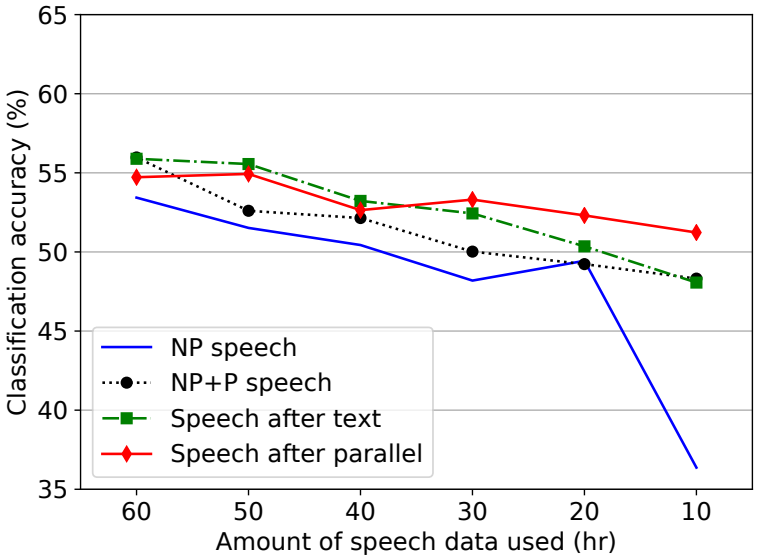


Figure 5.2: Classification accuracy versus the amount of non-parallel (NP) speech data when inputs are MFCCs

amounts of NP speech data (NP speech). Next, we combine the NP speech data with the speech portion of the parallel (P) data and train the speech branch on that set (NP+P speech). As the amount of data is larger in this situation, we achieve higher accuracy than the baseline. In multiview training, we first train the rich-resource text branch with NP data. We then freeze the classifier and train speech encoder on non-parallel speech data ("Speech after text" corresponds to the model at the end of Step 3 in Algorithm 1). As seen from the figure, pretraining the classifier on text and then learning the speech encoder on NP data performs better than training the speech model on NP+P data. We then fine-tune both text and speech branches using the limited amount of parallel text and speech ("Speech after parallel" corresponds to the model at the end of Step 5 in Algorithm 1). For the cases where we have more than 30 hours of speech, fine-tuning step does not bring any benefit. However, when we have less than 30 hours of speech, fine-tuning with parallel data improves the accuracy as compared to "Speech after text". In the fine-tuning stage we adjust both the encoders and the classifier whereas in "Speech after text", we only learn the encoder with classifier fixed.

Since multiview system allows us to test the system using unimodal data, we also report the text-only performance of the systems. Table 5.1 shows the classification accuracy of both speech and text branch after all training steps.

Table 5.1: Amount of non-parallel data (hr) to pretrain the branches and the accuracy of the text-only, speech-only and ASR-text based testing of the multiview model for the MFCC-based setup

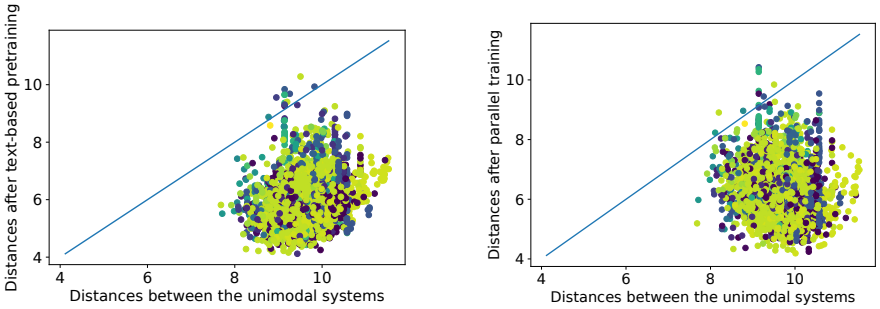
Training condition (in hr)			Test Accuracy		
Text	Speech	Parallel	Text	Speech	ASR-text
60	60	14.5	0.675	0.547	0.541
70	50	14.5	0.685	0.549	0.548
80	40	14.5	0.679	0.526	0.552
90	30	14.5	0.673	0.533	0.539
100	20	14.5	0.677	0.523	0.546
110	10	14.5	0.654	0.512	0.536

The speech accuracies in the table correspond to "Speech after parallel" curve in Fig. 5.2. Although training is performed on true transcript text, in practical scenarios we usually do not have the true text during test time but only ASR outputs. Therefore, we also show the results of testing the text branch with ASR-based text. We see how the mismatch between noisy and clean text affects the classification accuracy. We see that although true-text based testing gives above 65% performance, ASR-text based testing lowers the accuracy to that of the speech-only testing. Another disadvantage of ASR based testing is that it requires a language model in addition to an acoustic model whereas in the speech-only E2E classification, all we need is the acoustic features.

When we compare "NP speech" and "Speech after parallel" setups, for the low-resource case, we get 5-40% relative improvement in accuracy after fine-tuning with parallel data. The gain reaches to 40% (0.363 to 0.512) when we have only 10 hours of non-parallel speech at the beginning.

For the conditions achieving 40% relative improvement, which is the 10 hours of non-parallel speech scenario, we plot the distance between the text and speech embeddings to see if the proposed approach can tie them together using a shared classifier. If the hypothesis holds, then the distances after training should be smaller than the distance of the unimodal systems. As shown in Fig. 5.3, after applying either "Speech after text" or the "Speech after parallel" method, we get smaller distances between embeddings as they are mostly below the diagonal.

These results confirm several hypotheses. First, simple acoustic features



(a) Speech after text (b) Speech after parallel

Figure 5.3: Distance comparison between text and speech embeddings before and after multiview training

are harder to classify than text embeddings such as BERT. Second, although a text-based system works well if tested on true text, in practice we do not have access to that information and hence need to resort to ASR-based noisy text which deteriorates the results to the level of speech-only testing. Third, non-parallel text data can be used to guide learning speech encodings and it helps improve speech-only performance. Although we do not have the state-of-the-art results on the text branch [129], we can still improve the speech-only performance in the proposed multiview architecture. Our speech-only performance on the other hand achieves better than the best speech-only system reported in [61], which is at 38.9%.

As discussed in [61], even if two sentences are the same, depending on the context, the output classes can change in the dialog. Therefore, one method to further improve the speech-only performance is to use the context or the history of the dialog acts while making class decisions. Our initial experiments on using context, not presented in this dissertation, show that we can improve the accuracy further.

Another way of increasing the performance is to improve the speech features fed into the system. Note that text representations come from a pretrained BERT model; however, in the first set of experiments, speech features were MFCCs. Even though ASR text-based testing performs poorly, in the cases where we have access to an neural network based acoustic model, we can utilize it as a feature extractor. In the second set of experiments, we took an ASR model trained on the Switchboard dataset [130], and extracted speech features from the LSTM output of that acoustic model. We then repeated

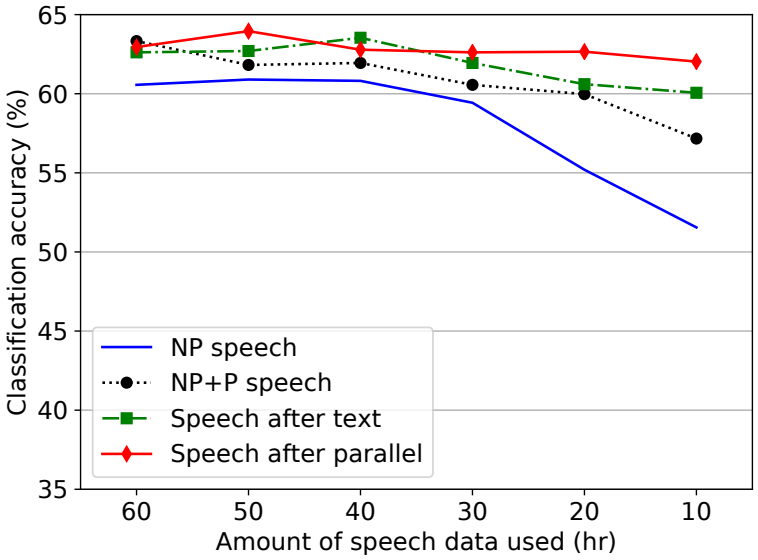


Figure 5.4: Classification accuracy versus the amount of non-parallel (NP) speech data when inputs are ASR based embeddings

the first set of experiments using these ASR-based speech features. As shown in Fig. 5.4, when we have sufficient amount of speech data, the unimodal speech-only training achieves above 60% accuracy. Our observations from the first experiments still hold for this case; i.e., text-based pretraining of the classifier and then learning the speech encoder (Speech after text) helps improve the performance and in the very low-resource case (less than 30 hours), additional fine-tuning (Speech after parallel) with the parallel data helps further increase the accuracy.

In Table 5.2, we report the true text and ASR-text based testing of the multiview model for the second set of experiments performed on ASR-based speech embeddings. In terms of the results, the major difference between the previous experiment and the current one is that here, speech-only results approach to the true text based performance and they are significantly better than ASR-text based testing. This shows that we can achieve better performance than an ASR+NLP system with speech-only training when ASR-based speech embeddings are used.

When we compare "NP speech" and "Speech after parallel" setups, for the low-resource case, we get between 5% and 20% relative improvement in accuracy after fine-tuning with parallel data. The largest gain is observed when we have 10 hours of non-parallel speech data (0.516 to 0.620). Although the

Table 5.2: Amount of non-parallel data (hr) to pretrain the branches and the accuracy of the text-only, speech-only and ASR-text based testing of the multiview model for the ASR embedding-based setup

Training condition (in hr)			Test Accuracy		
Text	Speech	Parallel	Text	Speech	ASR-text
60	60	14.5	0.677	0.630	0.549
70	50	14.5	0.688	0.640	0.549
80	40	14.5	0.672	0.628	0.535
90	30	14.5	0.681	0.626	0.558
100	20	14.5	0.682	0.627	0.556
110	10	14.5	0.672	0.620	0.543

relative improvements are not as large as the first experiments, the absolute accuracies are much better in this case. If we compare ASR text-based testing to the speech-only testing case, we achieve about 15% improvement in accuracy (roughly from 0.55 to 0.63).

We also performed multiview experiments on the ATIS speech-to-intent classification task. Table 5.3 shows the text-based accuracy, speech accuracy after multiview training and speech accuracy after parallel training. We compare three models: the first one is a speech-only model mapping speech utterances into intent with a classifier, the second one is the multiview model on the original dataset which has parallel text and speech, and the third model is also a multiview model but trained on non-parallel text and speech. In the last one, non-parallel text is obtained by augmenting the text data using a template structure. We identified the list of airports, cities and dates in the dataset, then for each text sentence in the dataset we checked if one of these fields exists in that sentence; if that is the case, we generated a copy of the sentence by replacing the name with one of the available ones without changing the label. For instance, given an utterance such as "I would like to fly from New York to Boston.", we generated an additional sentence such as "I would like to fly from New York to Chicago." as Chicago is one of the possible city names in the dataset. Note that this does not affect the label, both of these sentences have the "flight" intent.

As shown in Table 5.3, multiview training even without any text augmentation helps improve the speech-to-intent classification accuracy from 67% to 76.5%. If we use augmented text for multiview training, we observe that we

Table 5.3: Multiview training experiments on the ATIS speech-to-intent dataset, numbers denote the accuracy

Model	Text	Speech	Speech after Parallel
Speech-only	-	0.670	-
Multiview	0.861	0.765	0.733
Multiview with text augmentation	0.940	0.796	0.805

further increase the speech performance to 79.6%. If we also perform parallel training after this, then the accuracy becomes 80.5%.

With the experiments on two datasets, we showed that we can leverage non-parallel text data to help learn better speech embeddings in the speech-to-dialog act and speech-to-intent classification problems. We also showed that phonetically-aware speech embeddings from an ASR system can further help increase the final speech-to-concept performance.

CHAPTER 6

FAIR SPOKEN LANGUAGE UNDERSTANDING WITH DEEP F-MEASURE

In this chapter, we will propose an empirical method for optimizing the F_θ-measure in a DNN-based system. We will demonstrate the performance on both benchmark socio-economic datasets and on speech-to-intent and speech-to-image object tasks. This chapter is mainly based on our previous work in [131].

6.1 Introduction

Many machine learning datasets have a label imbalance or dataset bias problem. In many cases, either data is harder to collect for certain classes or the data collection phase is biased itself such that bias is introduced to the collected dataset. Typical training algorithms, optimized in order to minimize error, tend to do so by exacerbating bias, e.g., by providing higher recall and precision to the majority class than to minority classes. Therefore, the label imbalance problem raises the concern about fairness of machine learning systems in general [4, 84, 132]. SLU problems often suffer from label imbalance, in ways that may hide important errors from the designers of SLU systems.

Consider an SLU dataset such as Air Traffic Information Systems (ATIS) [133] and the speech-to-intent detection problem on this dataset. About 75% of the dataset carries the intent of searching for a flight, while conversely, some minority intent classes are represented by only a single training example; this is a severe label imbalance problem. Suppose that we train a model without any concerns about fairness or imbalance. The model will very likely learn to output the "flight" intent all the time, yielding an accuracy of 75% which is not low and could be acceptable depending on the application. Considering that there are roughly 30 classes in the whole dataset, one class will have a recall of 1.0 and precision of 0.75, and the remaining 29 classes

will have both recall and precision of 0.0. In such a scenario, the F-measure, which is a harmonic average of precision and recall, will be 0.86 for the most common class and 0.0 for the rest, giving an average of 0.03 which is not acceptable in many cases.

In this chapter, our goal is to design a loss function to maximize the F-measure instead of the accuracy for DNNs. Our methods are tested on two standard socioeconomic classification problems from the literature on fairness (the UCI [134] Adult [135] and Communities and Crime [136] tasks), and on two SLU tasks (intent classification in ATIS, and detection of the named object in spoken captions that name only one object from the Speech-COCO dataset [77]). On the SLU tasks, we perform E2E SLU, i.e., we directly map speech input to the labels instead of performing ASR followed by NLP. We pose the SLU problems as multi-class classification tasks and use the softmax output from the DNN, making it possible to apply the same optimization criterion to both the socioeconomic and SLU learning problems. We approximate the F-measure with a differentiable function of the softmax activations so that we can use the standard backpropagation algorithm [137] to train the DNN.

6.2 Deep F-measure Maximization

As mentioned in Chapter 2, F_θ measure is defined as

$$F_\theta = \frac{(1+\theta^2)TP}{\theta^2(TP+FN)+(TP+FP)} \tag{6.1}$$

for binary classification. In the multi-class case, we focus on the average per-class F-measure:

$$F_\theta = \frac{1}{K}\sum_{k=1}^{K}\frac{(1+\theta^2)TP(k)}{\theta^2 N_k + (TP(k)+FP(k))}, \tag{6.2}$$

where N_k term corresponds to $(TP(k)+FN(k))$.

6.2.1 Empirical Optimization of F_θ

Earlier works on F_θ-measure have focused on learning a threshold for making a decision for the binary classification problem. On the other hand, in the case of multi-class classification with DNNs, the class decision is made by taking the softmax at the output layer and then by choosing the class with the highest softmax activation. Therefore, in F_θ maximization with neural networks, we do not aim at identifying the threshold but designing a loss function that is differentiable so that we can use the backpropagation method to learn the DNN model parameters.

Equation (6.2) contains counting which is expressed using indicator functions that are not differentiable. For example, given that the softmax activations for the n^{th} data point, or token, are $\hat{y}_n(k),\ k = 1, 2, \cdots, K$ and that y_n is the one-hot representation of the true label, the number of true positives for a certain class k is written as

$$TP(k) = \sum_n \mathbf{1}[\arg\max y_n = k \wedge \arg\max \hat{y}_n = k], \tag{6.3}$$

where the indicator function $\mathbf{1}$ is not differentiable. Therefore, we need a differentiable approximation for F_θ. To achieve this, instead of the hard counts, we use the soft counts which are obtained from the softmax activations. To make the largest activations equal to 1, we do the following normalization on the activations for each token:

$$\hat{y}'_n = \frac{\hat{y}_n}{\max_k \hat{y}_n(k)}. \tag{6.4}$$

Using these soft counts, we approximate the terms in Eq. (6.2) as

$$TP(k) \approx \sum_{n \in S_k} \hat{y}'_n(k) \tag{6.5}$$

$$TP(k) + FP(k) \approx \sum_{n \in S} \hat{y}'_n(k), \tag{6.6}$$

where S_k denotes the set of indices for data tokens with label k and S is the set of all indices in the dataset. We do not approximate N_k as it is determined directly from the dataset. Thus, our loss function becomes the

negative of the approximate F_θ:

$$\mathcal{L} = -\frac{1}{K}\sum_{k=1}^{K}\frac{(1+\theta^2)\sum_{n\in S_k}\hat{y}'_n(k)}{\theta^2 N_k + \sum_{n\in S}\hat{y}'_n(k)}. \tag{6.7}$$

Since $\hat{y}'_n$ is a differentiable function of $\hat{y}_n$, it is also differentiable with respect to the DNN model parameters. Hence, we can learn the network weights by backpropagating the derivatives of the loss function in Eq. (6.7). The loss function in Eq. (6.7) is not specific to fully-connected neural networks but can be used for any neural network with a softmax output layer.

In the approximations given in Eqs. (6.5) and (6.6), instead of $\hat{y}'$, we could have used $\hat{y}$ directly, or we could have computed the softmax by first scaling the pre-softmax activations by a constant to increase the sharpness of the final activations. However, in our experiments, we saw that the approximations proposed in the equations above performed the best.

6.3 Experiments

In this section, we will describe two sets of experiments. Although our main focus will be on dealing with dataset bias in SLU systems, the first set of experiments will be on smaller datasets for non-speech, binary classification tasks. These are usually used as benchmark tasks as they reflect some societal bias. The second set of experiments will be on speech-to-intent and speech-to-concept classification which are both multi-class classification tasks. Details of the models and the results will be presented in the following subsections.

6.3.1 Experiments on Socioeconomic Data

The first set of experiments is performed on non-speech tasks. The goal here is to show whether the proposed method is providing any gains as compared to cross-entropy based training. Since the dataset bias is usually discussed in the realm of socioeconomic data with certain protected attributes such as race, gender, age-group etc., we first want to investigate whether we achieve an improvement in these tasks. For this task, we use two datasets from the UCI repository [134], namely, Adult [135] and Communities and Crime [136].

Table 6.1: Binary classification performance on two UCI datasets

Data	Loss	Prec	Rec	Micro-F_1	Avg-F_1	Accu.
Adult	xent	0.7977	0.6193	0.6973	0.6389	0.8085
	deepF	0.8196	0.6170	0.7040	0.6361	0.8107
C&C	xent	0.7422	0.7075	0.7245	0.7206	0.7940
	deepF	0.7541	0.7319	0.7428	0.7413	0.8040

In the Adult dataset, given the personal attributes (age, race, marital status, education level, etc.) of a person, the goal is to estimate whether the person has an income over \$50K/year. The majority class, i.e. individuals with income less than \$50K/year, comprises 76% of the data points. In the Communities and Crime (C&C) dataset, the goal is to detect if a community has a high crime rate where, as described in [138, 139], we define "high crime rate" to mean a crime rate above the 70^{th} percentile of the training dataset. The majority class, i.e., low crime-rate, comprises 70% of the samples.

Both the Adult and C&C tasks are two-class problems, for which a standard F-measure is well-defined. Our interest is the maximization of a multi-class F-measure; therefore, the F-measures of both majority and minority classes are first computed, and then averaged as shown in Eq. (6.7).

In both tasks, we use fully-connected neural networks with 16 units per layer. The numbers of layers are 7 and 4 for the Adult, and C&C datasets, respectively. The output is a softmax layer with 2 units. As a baseline, we use the models trained with cross-entropy loss and compare them to models trained by the proposed deep F_θ loss. Table 6.1 shows the average precision, average recall, micro-F_1 and classification accuracy for both cross-entropy model (xent) and the proposed model (deepF) for both datasets where we take $\theta = 1$. For both datasets, we improve the micro-F_1 and accuracy. For the C&C dataset, we also see improvement in the average-F_1 score.

6.3.2 Experiments on Spoken Language Understanding

The second set of experiments is on speech related tasks. We investigate direct speech-to-meaning systems where instead of the conventional two-step process (ASR+NLP), our goal is to directly understand the speech signal in an E2E framework. For the SLU problem, we run experiments on two tasks:

Table 6.2: Number of classes and the frequency (in %) of the most frequent top-3 classes for ATIS and Speech-COCO datasets based on the training data

Data	#Classes	Top1	Top2	Top3
ATIS	29	73.7	8.5	5.1
Speech-COCO	80	22.6	3.5	3.1

speech-to-intent detection and speech-to-concept classification, both of which are multi-class classification problems. We work on the ATIS dataset [133] for the speech-to-intent task, where the intents are "searching for a flight", "getting airport information", "local transportation options", etc. There are 29 intents in the whole dataset 8 of which do not appear in the training set. For the speech-to-concept task, we use the Speech-COCO dataset [77]. This dataset consists of synthesized speech signals for the image captions in the MS-COCO dataset [140]. We define the task to be mapping the spoken image captions to the image label. There are 80 classes in the dataset.

In Table 6.2, we show the number of classes and the frequency of the most common three labels in both ATIS and Speech-COCO training sets. As shown in this table, the classes are highly imbalanced and we have dataset bias. Given these statistics, a model that always predicts the majority class will have 73.7% and 22.6% accuracy on the ATIS and Speech-COCO training datasets, respectively. If we compute the micro-F1 for such models, they will be 0.0293 for ATIS and 0.0046 for Speech-COCO, which are very low (less than 3%), and these numbers will get even lower for datasets with more classes. Especially, in the ATIS case, we see that relatively high accuracy does not necessarily mean a classifier that is fair to all classes.

E2E SLU has gained interest as a means to overcome the error propagation problem, in which speech transcription errors cause speech understanding errors [2, 71, 68, 69, 70, 123]. This work uses the speech branch of the multiview model described in [123] which consists of a BLSTM based encoder and a classifier with fully-connected layers (Fig. 6.1). Since our focus is on designing the loss function for F-measure maximization, we keep the DNN architecture otherwise identical to that used in [123], and use speech-only training instead of the multi-task training protocol described in [123]. For ATIS experiments, the model has a single BLSTM layer with 128 units and two fully-connected layers with 64 units each. For Speech-COCO experiments, the model has 2

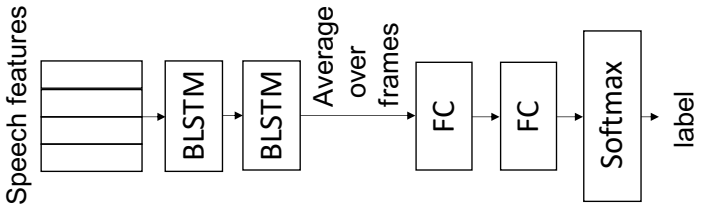


Figure 6.1: Our E2E SLU architecture based on [123]

Table 6.3: Multi-class classification performance (precision, recall, micro-F1, average-F1, accuracy and coverage) on E2E SLU problems for different models (M1: ReLU nonlinearity, M2: leaky ReLU nonlinearity)

		M1 - ReLU nonlinearity						M2 - leaky ReLU nonlinearity					
Data	Loss	Prec	Rec	Mic-F_1	Avg-F_1	Accu	C	Prec	Rec	Mic-F_1	Avg-F_1	Accu	C
ATIS	xent	0.0244	0.0345	0.0286	0.0286	0.7772	1	0.0313	0.0362	0.0336	0.0332	0.6697	2
	deepF	0.0520	0.0554	0.0536	0.0516	0.6484	4	0.1054	0.0936	0.0991	0.0947	0.7447	5
COCO	xent	0.1992	0.2268	0.2121	0.1956	0.3538	50	0.3876	0.3716	0.3794	0.3509	0.4473	74
	deepF	0.2539	0.3137	0.2807	0.2676	0.3264	79	0.3927	0.3994	0.3960	0.3895	0.4439	79

BLSTM layers with 128 units each and two fully-connected hidden layers with 128 and 64 nodes. The dataset comes with train and validation splits; we reserve 25% of the training subset as our development set. In both cases, we experiment with ReLU and leaky ReLU non-linearity for the fully-connected layers, we set the learning rate to 0.001, and we use Adam optimizer.

In Table 6.3, we show the average precision, average recall, micro-F_1, average-F_1, accuracy and coverage. We define the coverage as the number of classes with non-zero recall. This is an indicator of fairness as it highlights the very low number of classes that have non-zero recall under a standard cross-entropy training paradigm. We report the results on both ATIS and Speech-COCO datasets. Training with cross-entropy loss is compared to training with the proposed F_θ measure (with $\theta = 1$). We first experiment with model 1 (M1) that has ReLU non-linearity. For both datasets, we see that deep F-measure maximization (deepF) results in higher micro-F_1 and average-F_1 as compared to the cross-entropy (xent) model. In both cases, we also see that we increase the coverage significantly. Especially, on the ATIS dataset, we see that the cross-entropy model only outputs the majority class label. On the other hand, the deepF model has a coverage of 4 which shows that it is able to output labels from different classes. On the Speech-COCO dataset, with the deepF model, we cover almost all classes (79 out of 80). However, we also observe that there is a trade-off between coverage and ac-

curacy. While trying to cover different classes, the model misses some of the majority class data points which leads to slightly lower accuracy as compared to the cross-entropy model. This is an expected outcome as the deep F-measure optimization aims at achieving better F-measure without paying attention to the overall accuracy. If our goal is fairness, and if the difference in accuracy is not large, deepF may still be the preferred approach. When we trained M1 for larger θ (more emphasis on recall), we saw that ReLU neurons start to die and hence lead to the degenerate solution, i.e., outputting the majority class label. Therefore, we also perform experiments with leaky ReLU (model 2, M2). With M2, we observe better baselines with the cross-entropy objective. However, our previous conclusions still hold: deepF leads to higher F-measure and increased coverage.

In Fig. 6.2, we show the average-F_θ and micro-F_1 obtained from M2 for ATIS and Speech-COCO datasets, for different values of θ. Note that in the case of cross-entropy training, we only train a single model, then compute its F_θ for different values of θ. On the other hand, we train a model for each θ in the case of deep F-measure maximization. The cross-entropy system is trained for 25 epochs. The deep-F system is trained for 15 epochs using cross-entropy, then for 10 epochs using the F_θ measure.

Results on the ATIS dataset (lower half of the results in Fig. 6.2) show that the proposed deep F-measure maximization approach leads to 6-8% absolutely higher micro-F_1 and average-F_θ as compared to the cross-entropy model for a wide range of θ. By comparing M2 results in Table 6.3 to Fig. 6.2, it is possible to compare the sizes of the improvements in coverage (about 3-fold improvement at $\theta = 1$) and in F_1. Micro-F_1 improves by a factor of 2.9 at $\theta = 1$, and by a factor of 3.2 at $\theta = 4$ (from 0.0359 to 0.1161). These results suggest that increasing coverage has a large (up to 8% absolute) effect on the micro-F_1.

As shown in upper half of the Fig. 6.2, for the Speech-COCO dataset, F-measures are around 35-40%. On this dataset, deep F-measure maximization still performs better (up to 5% absolute) than the cross-entropy loss when $\theta < 4$ and there is not a significant difference in the F-measure for different θ. However, when $\theta \geq 4$, the performance starts to fall below the cross-entropy model. Still, if we look at the coverage for these models, we see that it is 79 which is higher than that of the cross-entropy model. This means that we have nonzero recall for more classes but the individual F-measures per class

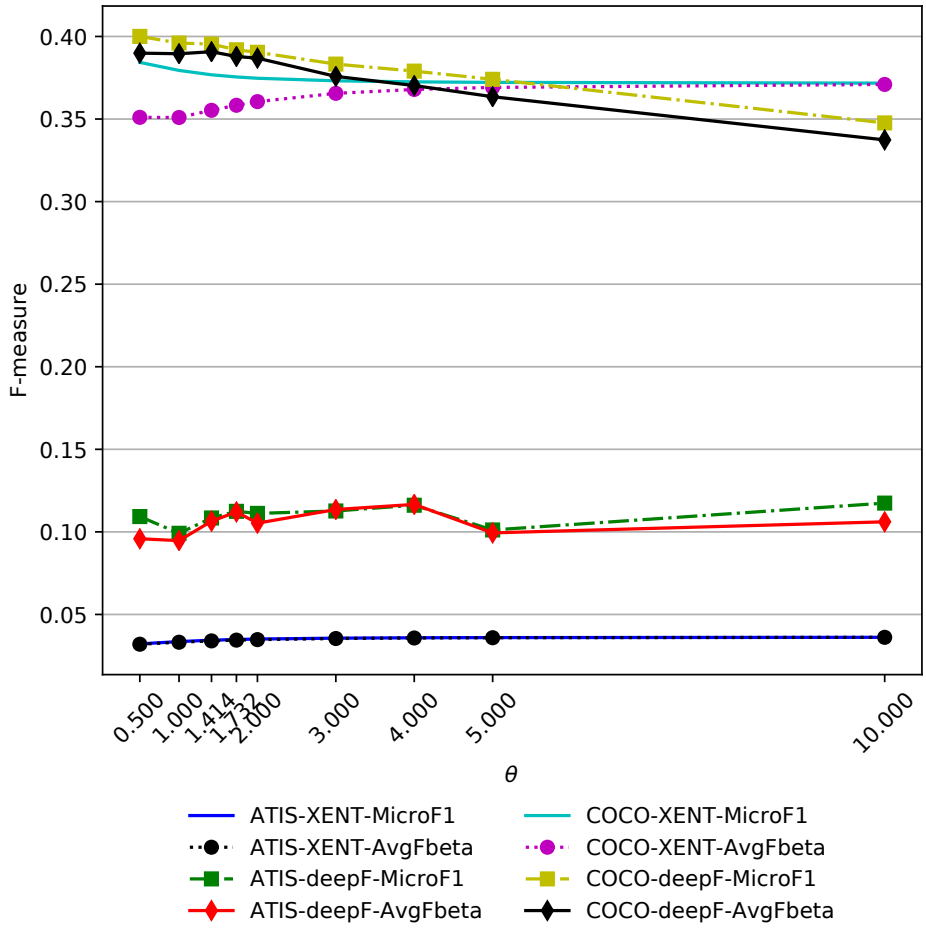


Figure 6.2: Micro-F_1 and average-F_θ values on the ATIS and Speech-COCO datasets for different θ after training with cross-entropy (XENT) or deep F-measure (deepF) losses

are, on average, lower than their cross-entropy counterparts.

CHAPTER 7

DISCUSSION

As mentioned in the Introduction (Chapter 1), learning speech embeddings is a very general topic and we cannot cover all the aspects in a single thesis. It is possible to learn the embeddings in various ways such as assuming a distance measure and mapping the inputs to a space where this distance measure has a physical meaning, using a structured model that is crafted towards extracting a certain type of knowledge (e.g., speaker characteristics), or designing an objective function such that the model, and hence the embeddings from that model, would be meaningful. In this dissertation, we particularly followed the last two approaches for each of the problems at hand, namely ASR and SLU. In each case, we provided a model and an objective, which led to four standalone projects that we discussed in Chapters 3-6.

This chapter will focus on possible impacts of the proposed models or algorithms in various domains. Once deployed on a real ASR or an SLU system, the major impact will be the increase in user satisfaction from the human-computer interaction (HCI) system. In addition, the fairness component will also have societal and psychological impacts on the users. We will then discuss future directions in which these studies can possibly lead.

7.1 Research Impact

The joint speaker change detection and adaptation method proposed in Chapter 3 could be utilized in streaming ASR systems. Conventional systems perform these two tasks separately which causes delay in the output. However, when there are frequent speaker changes in the input signal or in a dialog in which one speaker speaks only for short intervals, it would be faster to process the data in one pass rather than splitting the data into small single speaker regions. As shown in the experimental results, this method also

reduces the WER which increases the user satisfaction of the ASR system which could be deployed into a mobile device or a home assistant such as Amazon Alexa or Google Home.

In Chapter 3, we showed that we reduce the WER using the proposed auxiliary network in a hybrid DNN-HMM model but the proposal does not assume the type of ASR, i.e. it is equally applicable to the E2E ASR systems. In later experiments, we tried using the auxiliary network in the DeepSpeech based E2E ASR model of Chapter 4, and we observed that the auxiliary model still lowers the WER. This supports the claim that the auxiliary network idea is agnostic to the exact network architecture. On the other hand, when we evaluated the speaker level WERs of this model on the CORAAL dataset, we observed that the WER reduction for female speakers was greater than that for the male speakers. This resulted in a larger standard deviation of WER across test speakers as compared to the unadapted baseline model. This shows that fairness against speaker differences is not completely achieved by the auxiliary model and supports the necessity of a fair algorithm for speaker adaptation.

As discussed in Chapter 4, it is possible to approach the adaptation from a fair machine learning perspective. This means that we can reduce the error rate gap between genders, races, or dialects and improve inclusiveness of ASR systems. This in turn leads to covering more users with better ASR quality which promotes equality, diversity and authenticity. Especially, the last point has social and economic consequences. For example, there are studies showing that the African American community tends to code-switch to "proper" or standard English in formal settings or in their work life in order to advance their careers. There are discussions on the implications of this forced code-switching such as change in self-image (self-confidence, identity, being respected by others) which can lead to mental issues, as well as economic impacts such as not getting the raise or promotion they deserve. Given that standard ASR systems are usually trained on American English, African American users may not find the commercial ASR systems useful for them. However, if we can make the ASR model fair, we can alleviate some of these problems.

The equity theory proposed by Adams [141] suggests that the perceived fairness of people depends on how their outcome-to-input ratio compares with their perception of other people's ratios. This theory, which is im-

portant for social psychology, also affects how people feel, think and behave [142]. Consider the following scenario: Two females, one black and one white, want to test a device that supports voice commands; they speak the same sentence but in the first trial the black female's speech is not understood by the machine and she tries several times until she achieves her goal. On the other hand, for the same voice command, one iteration is sufficient for the white speaker. From the black person's perspective, she got the same outcome as the white female at the expense of more input, i.e. more repetitions. Hence the perceived outcome-to-input ratio is lower for the black female as compared to her white friend. This causes perceived unfairness and dissatisfaction from the service.

As summarized in [143], emotional reactions to unfairness are closely related to stress and can therefore lead to physical changes as part of the general stress response such as high blood pressure. Negative emotions such as anger are also dominant responses to unfairness. Sustained negative feelings and stress response can eventually lead to other physical and mental problems in the long term. We think that having fair machine learning systems can prevent a (small) part of injustice and can contribute to the well-being of individuals and minorities.

As for the theory of counterfactual training, whether there must be a trade-off between accuracy and fairness is also under discussion; these are especially hard to prove mathematically for a complex problem such as ASR. Although we empirically observed that there is usually a trade-off for two solutions (CTC matching and posterior matching), in the logit matching experiments, we were able to find a set of hyperparameters where we can have both low CER and low gender disparity as compared to a baseline model. This might suggest that the trade-off is not a must but further evaluations might be necessary.

It is usually harder to obtain speech recordings as compared to text data as speech carries identity information which causes privacy concerns. Therefore, we think that the method proposed in Chapter 5 will be helpful for languages that have limited amount of speech but large amounts of text data. The reason for focusing on E2E SLU was that we wanted to bypass the ASR+NLP approach. In Chapter 5, we showed the performance gap between ground truth text and ASR text based performances which suggests the importance of the E2E approach. Such an approach would possibly reduce the latency

as we do not require a full ASR decode. This can lead to faster interaction and shorter wait time to get a response from a smart device and also lead to smoother or more natural verbal communication between humans and machines.

Even though we proposed the multi-view training for the text and speech modalities, in a separate study [144] we showed that we can also use this idea with video and speech modalities for the purpose of audio-visual cross-modal speaker verification, i.e. matching faces with voices. In that study, we showed that we can achieve accuracy comparable to the existing methods on the VoxCeleb1 and VoxCeleb2 datasets. Hence the main training framework is applicable to different modalities.

In Chapter 6, although our focus was on SLU, the proposed method, namely the deep F-measure, is a general technique that can be used for an arbitrary network performing multi-class classification using a softmax output layer. We supported this claim by showing results on socio-economic datasets as well as speech. We think that deep-F-measure would be helpful for the cases where the training dataset is highly imbalanced and it is hard either to collect additional data for certain classes or to augment the dataset.

The foregoing discussion is related to responsibility from both the researcher's and user's perspective. In order to conduct responsible research, one dimension is to be inclusive as described in [145]. The chapters on fairness (Chapters 4 and 6) were part of this effort. Although we discussed the positive sides of the proposed research such as increased user satisfaction from an HCI system, we should also warn the users about possible negative impacts. For instance, when the user enjoys the conversation with the machine, they will tend to use the device more frequently which could possibly lead to the general problems associated with overuse of digital devices. Some of these effects are psychological (distraction, expectation of instant gratification, narcissism), social (isolation, deficit in social skills) or physical (hearing and vision problems, neck strain, less active life) [146]. From the user's point of view, they should be aware of these possible outcomes and use digital devices responsibly.

7.2 Future Directions

In this section, we will provide possible future directions that can have a short or a long time-span. The short-term directions mainly involve straightforward extensions of the current experiments or the settings, whereas the long-term ones correspond to more open problems and research questions that would be more involved.

7.2.1 Short-term Directions

One obvious extension of the auxiliary network proposed in Chapter 3 is to try the model on E2E ASR systems. In our fairness experiments, we started investigating this in the case of DeepSpeech2 model and we indeed observed gains in the accuracy. In order to show the effectiveness in the case of E2E models, we may also want to experiment with larger datasets (e.g. larger than 1000 hours of speech). This dissertation has reported our experimentation with utterances containing a single change point. However, the model is flexible to handle multiple change points and further tests can be performed on multi-turn utterances.

Our starting point for speaker adaptation was the failure of ASR systems on non-native speech. Hence, it might be useful to test the proposed model on accented speech and provide modifications to the model if necessary.

In Chapter 4, we proposed the counterfactual training to reduce the gender gap, age group and education group gaps. However, the idea is applicable to other dimensions of the speakers such as race or socio-economic status. We have ongoing efforts to investigate these aspects, specifically the dialect difference. One challenge of these is to generate realistic counterfactuals in the case of speech.

To answer concerns regarding whether the regularization effect of the counterfactual loss functions will disappear in the case of large datasets could also be a subject of a study. These models can be trained on large speech corpora. We also have ongoing efforts on experimenting with stronger baselines trained on larger datasets.

In Chapter 5, the multiview model provided a way to learn text-like speech embeddings through a shared classifier. Due to limited amount of non-English SLU corpora, we performed all SLU experiments on English speech.

It would be useful to test the proposal on other languages where we could have more text data than speech.

In the case of Deep-F training, one analysis that could be important for supporting our claims is to measure the performance gain with respect to the skewness of the label distribution. This type of analysis may give better insights about the accuracy vs. fairness trade-offs.

7.2.2 Long-term Directions

In Section 7.1, one motivation for simultaneously performing speaker change detection and speaker adaptation is given as the ability to use it in online systems. One caveat is that in its current form, the auxiliary model needs to wait until the end of the utterance so that it can make comparisons between speech segments; hence, there will still be some delay in processing. However, by reducing the look-ahead duration for the purposes of comparison, the delay might be reduced. Such a study can bridge the gap between the proposed model and a real-life implementation.

Speaker variability affects most of the speech applications, and SLU is one of those. Therefore, it is possible to introduce the speaker adaptation method of this work to the SLU task. Especially, the auxiliary model is an unsupervised adaptation method and can be applied to the SLU datasets, which usually do not contain speaker labels, to improve their accuracy. Such a study will also combine the two application areas discussed in this dissertation.

In this dissertation, we focused on learning embeddings geared towards specific applications. It would be interesting to see if we can learn generic speech embeddings from a single model that can perform ASR and SLU simultaneously depending on the output structure. There are some studies making use of multi-tasking, such as [147, 148], but they are not yet as popular as the task-specific systems. This holistic way of learning speech embeddings is a general line of research that can be handled in the future.

In our applications, our focus was mainly on English datasets. However, to be inclusive, it is important to provide such tools for non-English datasets. This fact proposes learning possibly multi-lingual speech embeddings. Especially, learning multi-lingual semantic speech embeddings would be a large

and interesting research question.

The counterfactual speaker adaptation method proposed in Chapter 4 relies on the generation of counterfactuals, i.e. voice conversion or speech generation for non-existing speakers. Although there are several proposals for time sequence generation or zero-shot voice conversion, these are still open issues that can be addressed in the future.

In Chapter 5, we proposed the multiview model to learn text-like speech embeddings through a shared classifier from non-parallel data. However, we could have imposed additional constraints on the embeddings to map speech and text to the same space. For example, there are cross-modal studies in image and voice or image and text. Those methods usually require parallel data but coming up with solutions without this constraint is a direction for future research.

Due to the shared classifier structure, it might be argued that the multiview model requires having similar labels for different modalities. The problem of having an open-domain evaluation, i.e., handling unseen classes in the classifier, is currently an open problem and subject to future research.

In our SLU studies, we tackled the problems that require classifying the spoken sentence into a label such as the intent. However, SLU is a much more general problem than utterance classification; e.g., consider the slot-filling problem. It might be interesting to see if the embeddings learned from a model such as the proposed multi-view model can also be useful for slot-filling tasks.

The deep F-measure proposed in Chapter 6 is based on the soft-counts idea which can be improved by finding other differentiable ways of obtaining the statistics such as TP, FP, TN and FN based on the ground truth. In addition, the current proposal was highly empirical. It might be useful to show the theoretical reasons underlying this mechanism or to provide an alternative differentiable F-measure approximation derived from theory. This could provide theoretical guarantees or bounds that could make it easier to compare with the conventional objective functions such as cross-entropy.

One general note is that although we did not use speaker labels in Chapter 3, Chapters 4-6 assume that labels corresponding to the task are available. Therefore, we mainly worked on supervised learning of embeddings. Given the fact that data collection and labeling are expensive tasks, semi-supervised or unsupervised learning of these embeddings is a major direction for future

research. However, this could bring complications especially in the case of fairness because some fairness criteria depend on conditioning on the true label which will not be available in an unsupervised setting. This may lead to proposals for fair and unsupervised learning. At this point, the fairness of unsupervised systems is not well-defined and merits investigation.

CHAPTER 8

CONCLUSIONS

In this dissertation, we discussed two application areas of speech processing, namely ASR and SLU. We mainly focused on speaker adaptation for ASR and E2E SLU. Since we are trying to map speech input to a set of outputs through neural networks, the overarching theme of this work was learning speech embeddings. We also discussed that there are concerns for fairness in these applications, and we proposed some novel ways of learning fair speech embeddings in these problems.

In Chapter 3, we introduced an auxiliary network for speaker adaptation and then combined it with the speaker attention mechanism in order to simultaneously detect speaker changes and adapt to the speaker. We showed that even if we did not use the change point information and speaker labels during training, the model was able to learn to detect speaker changes. This gave us a way to process the multi-speaker input on-the-fly instead of performing change detection followed by speaker adaptation.

In Chapter 4, we proposed a speaker adaptation approach based on the counterfactual fairness paradigm. Specifically, we derived the individualized counterfactual equal odds and equal opportunity loss functions for E2E ASR. The former led to the matching of the character posterior probabilities of the real and the counterfactual inputs given the ground truth sequence, and the latter led to the matching of the CTC losses. We observed that there is usually a trade-off between accuracy and fairness of the ASR system. Still, in the logit matching experiments, we observed that it is possible to reduce the standard deviation of the CER for the test speakers as well as the overall CER.

In Chapter 5, we proposed a multi-view approach that uses a shared classifier on top of separate speech and text encoders. The training algorithm allowed us to use non-parallel text to first train the text branch which later guided learning of the speech embeddings. We showed that if we have a large

amount of text and limited amount of speech, we can improve the speech-to-concept performance in an E2E SLU system.

In Chapter 6, we observed that in the case of highly label imbalanced datasets, optimizing for prediction accuracy is not fair to the minority classes. To solve this problem, we proposed that we should maximize the F-measure instead of accuracy. Then the problem became finding a differentiable approximation to the F-measure so that we can use it with backpropagation. We proposed the soft counts idea to calculate this approximation, and we showed that we increase the number of classes with non-zero recall as compared to a standard cross-entropy based training.

In Chapter 7, we discussed several implications of the above-mentioned studies and provided directions for possible future research. In terms of impact, the direct implication is an increase in user satisfaction from verbal HCI systems such as home assistant devices. Learning fair embeddings also has a more general impact on society and the well-being of individuals. In terms of future research, we discussed some straightforward extensions of the current studies and also suggested learning speech embeddings in a semi-supervised or unsupervised manner.

There have been many studies on speaker adaptation and cross-modal training; hence, our proposals may just be small additional steps in these areas. We believe that the main novelty of this work was learning fair speech embeddings. Therefore, we hope that the fairness aspect of this research effort invites other researchers to think about this social aspect of speech processing.

REFERENCES

[1] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19 143–19 165, 2019.

[2] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 720–726.

[3] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning*. Online, 2019. [Online]. Available: http://www.fairmlbook.org/

[4] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big Data*, vol. 5, no. 2, pp. 153–163, 2017.

[5] K. Holstein, J. Wortman Vaughan, H. Daumé III, M. Dudik, and H. Wallach, "Improving fairness in machine learning systems: What do industry practitioners need?" in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–16.

[6] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[7] R. Tatman, "Gender and dialect bias in YouTube's automatic captions," in *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, 2017, pp. 53–59.

[8] M. Garnerin, S. Rossato, and L. Besacier, "Gender representation in French broadcast corpora and its impact on ASR performance," in *Proceedings of the 1st International Workshop on AI for Smart TV Content Production, Access and Delivery*, 2019, pp. 3–9.

[9] A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel, "Racial disparities in automated speech recognition," *Proceedings of the National Academy of Sciences*, vol. 117, no. 14, pp. 7684–7689, 2020.

[10] "Aspire chain model," https://kaldi-asr.org/models/m1.

[11] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *Advances in neural information processing systems*, 2017, pp. 4066–4076.

[12] S. P. Rath, D. Povey, K. Veselỳ, and J. Cernockỳ, "Improved feature processing for deep neural networks." in *Proc. ISCA Interspeech*, 2013, pp. 109–113.

[13] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors." in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 55–59.

[14] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 171–176.

[15] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6359–6363.

[16] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7942–7946.

[17] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 11, pp. 1938–1949, 2015.

[18] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 12, pp. 1713–1725, 2014.

[19] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *Proc. ISCA Interspeech*, 2017, pp. 132–136.

[20] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gang, and B.-H. Juang, "Speaker-invariant training via adversarial learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5969–5973.

[21] M. Delcroix, S. Watanabe, A. Ogawa, S. Karita, and T. Nakatani, "Auxiliary feature based adaptation of end-to-end ASR systems," *Proc. ISCA Interspeech*, pp. 2444–2448, 2018.

[22] X. Cui, V. Goel, and G. Saon, "Embedding-based speaker adaptive training of deep neural networks," in *Proc. ISCA Interspeech*, 2017, pp. 122–126.

[23] L. Sarı and M. Hasegawa-Johnson, "Speaker adaptation with an auxiliary network," in *Machine Learning in Speech and Language Processing Workshop (MLSLP)*, 2018.

[24] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[25] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[26] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2011, pp. 24–29.

[27] M. Kitza, R. Schlüter, and H. Ney, "Comparison of BLSTM-layer-specific affine transformations for speaker adaptation," *Proc. ISCA Interspeech*, pp. 877–881, 2018.

[28] K. Veselỳ, S. Watanabe, K. Žmolíková, M. Karafiát, L. Burget, and J. H. Černockỳ, "Sequence summarizing neural network for speaker adaptation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5315–5319.

[29] L. Sarı, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised speaker adaptation using attention-based speaker memory for end-to-end ASR," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7384–7388.

[30] Z. Fan, J. Li, S. Zhou, and B. Xu, "Speaker-aware speech-transformer," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2019, pp. 222–229.

[31] C. Zhang and P. C. Woodland, "DNN speaker adaptation using parameterised sigmoid and ReLU hidden activation functions," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5300–5304.

[32] L. Sarı, S. Thomas, and M. Hasegawa-Johnson, "Learning speaker aware offsets for speaker adaptation of neural networks," in *Proc. ISCA Interspeech*, 2019, pp. 769–773.

[33] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[34] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. ISCA Interspeech*, 2017, pp. 999–1003.

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

[36] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 6645–6649.

[37] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[38] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[39] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Proc. ISCA Interspeech*, 2017, pp. 949–953.

[40] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 369–376.

[41] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Proc. Interspeech 2017*, 2017, pp. 3707–3711.

[42] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[43] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for English conversational speech recognition," in *Proc. ISCA Interspeech*, 2017, pp. 959–963.

[44] Z. Meng, Y. Gaur, J. Li, and Y. Gong, "Speaker adaptation for attention-based end-to-end speech recognition," in *Proc. ISCA Interspeech*, 2019, pp. 241–245.

[45] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA Speech Recognition Workshop*, 1997, pp. 97–99.

[46] S. S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, vol. 8, 1998, pp. 127–132.

[47] D. Liu and F. Kubala, "Fast speaker change detection for broadcast news transcription and indexing," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[48] A. K. Sarkar, D. Matrouf, P. M. Bousquet, and J.-F. Bonastre, "Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification," in *Proc. ISCA Interspeech*, 2012, pp. 2662–2665.

[49] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason, "I-vector based speaker recognition on short utterances," in *Proc. ISCA Interspeech*. International Speech Communication Association (ISCA), 2011, pp. 2341–2344.

[50] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 5, pp. 1505–1512, 2006.

[51] D. Dimitriadis and P. Fousek, "Developing on-line speaker diarization system," in *Proc. ISCA Interspeech*, 2017, pp. 2739–2743.

[52] D. Pelleg, A. W. Moore et al., "X-means: Extending k-means with efficient estimation of the number of clusters." in *Proc. ICML*, vol. 1, 2000, pp. 727–734.

[53] H. Bredin, "Tristounet: Triplet loss for speaker turn embedding," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5430–5434.

[54] A. Jati and P. Georgiou, "An unsupervised neural prediction framework for learning speaker embeddings using recurrent neural networks," *Proc. ISCA Interspeech*, pp. 1131–1135, 2018.

[55] R. Wang, M. Gu, L. Li, M. Xu, and T. F. Zheng, "Speaker segmentation using deep speaker vectors for fast speaker change scenarios," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5420–5424.

[56] N. Zeghidour, G. Synnaeve, N. Usunier, and E. Dupoux, "Joint learning of speaker and phonetic similarities with Siamese networks," in *Proc. ISCA Interspeech*, 2016, pp. 1295–1299.

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

[58] S. H. Yella, A. Stolcke, and M. Slaney, "Artificial neural network features for speaker diarization," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 402–406.

[59] R. Yin, H. Bredin, and C. Barras, "Speaker change detection in broadcast TV using bidirectional long short-term memory networks," in *Proc. ISCA Interspeech*. ISCA, 2017, pp. 3827–3831.

[60] M. Hrúz and Z. Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4945–4949.

[61] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational linguistics*, vol. 26, no. 3, pp. 339–373, 2000.

[62] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[63] P. Haffner, G. Tur, and J. H. Wright, "Optimizing SVMs for complex call classification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 2003, pp. I–I.

[64] D. Hakkani-Tür, G. Tür, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Proc. ISCA Interspeech*, 2016, pp. 715–719.

[65] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[66] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[67] Q. Chen, Z. Zhuo, and W. Wang, "BERT for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.

[68] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5754–5758.

[69] A. Caubrière, N. Tomashenko, A. Laurent, E. Morin, N. Camelin, and Y. Estève, "Curriculum-based transfer learning for an effective end-to-end spoken language understanding and domain portability," *arXiv preprint arXiv:1906.07601*, 2019.

[70] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *arXiv preprint arXiv:1904.03670*, 2019.

[71] Y. Qian, R. Ubale, V. Ramanaryanan, P. Lange, D. Suendermann-Oeft, K. Evanini, and E. Tsuprun, "Exploring ASR-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2017, pp. 569–576.

[72] Y.-P. Chen, R. Price, and S. Bangalore, "Spoken language understanding without speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6189–6193.

[73] J. Searle, *Expression and meaning: Studies in the theory of speech acts*. Cambridge University Press, 1979.

[74] J. L. Austin, *How to do things with words*. Oxford University Press, 1975.

[75] H. P. Grice, "Utterer's meaning and intentions," *The philosophical review*, vol. 78, no. 2, pp. 147–177, 1969.

[76] J. Margolis, "Meaning, speakers' intentions, and speech acts," *The Review of Metaphysics*, vol. 26, no. 4, pp. 681–695, 1973.

[77] W. Havard, L. Besacier, and O. Rosec, "Speech-COCO: 600k visually grounded spoken captions aligned to MSCOCO data set," *CoRR*, vol. abs/1707.08435, 2017. [Online]. Available: http://arxiv.org/abs/1707.08435

[78] A. Chouldechova and A. Roth, "The frontiers of fairness in machine learning," *arXiv preprint arXiv:1810.08810*, 2018.

[79] S. Verma and J. Rubin, "Fairness definitions explained," in *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*. IEEE, 2018, pp. 1–7.

[80] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Francisco, CA: Morgan Kaufman, 1988.

[81] P. Bickel, E. Hammel, and J. O'Connell, "Sex bias in graduate admissions: Data from Berkeley," *Science*, vol. 187, no. 4175, pp. 398–404, 1975.

[82] M. Srivastava, H. Heidari, and A. Krause, "Mathematical notions vs. human perception of fairness: A descriptive approach to fairness for machine learning," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[83] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012, pp. 214–226.

[84] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 3315–3323.

[85] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic decision making and the cost of fairness," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 797–806.

[86] N. Grgic-Hlaca, M. B. Zafar, K. P. Gummadi, and A. Weller, "The case for process fairness in learning: Feature selection for fair decision making," in *NIPS Symposium on Machine Learning and the Law*, vol. 1, 2016, p. 2.

[87] J. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent trade-offs in the fair determination of risk scores," *arXiv preprint arXiv:1609.05807*, 2016.

[88] S. Karthik et al., "The impossibility theorem of machine fairness–a causal perspective," *arXiv e-prints*, pp. arXiv–2007, 2020.

[89] R. Binns, "On the apparent conflict between individual and group fairness," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 514–524.

[90] S. Dutta, D. Wei, H. Yueksel, P.-Y. Chen, S. Liu, and K. Varshney, "Is there a trade-off between fairness and accuracy? A perspective using mismatched hypothesis testing," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2803–2813.

[91] J. Pearl, "Causal diagrams for empirical research," *Biometrika*, vol. 82, no. 4, pp. 669–688, 1995.

[92] C. Dwork and C. Ilvento, "Group fairness under composition," in *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency (FAT* 2018)*, 2018.

[93] S. Dash and A. Sharma, "Counterfactual generation and fairness evaluation using adversarially learned inference," *arXiv preprint arXiv:2009.08270*, 2020.

[94] J. Joo and K. Kärkkäinen, "Gender slopes: Counterfactual fairness for computer vision models by attribute manipulation," *arXiv preprint arXiv:2005.10430*, 2020.

[95] D. Kaushik, E. Hovy, and Z. C. Lipton, "Learning the difference that makes a difference with counterfactually-augmented data," *arXiv preprint arXiv:1909.12434*, 2019.

[96] P.-S. Huang, H. Zhang, R. Jiang, R. Stanforth, J. Welbl, J. Rae, V. Maini, D. Yogatama, and P. Kohli, "Reducing sentiment bias in language models via counterfactual evaluation," *arXiv preprint arXiv:1911.03064*, 2019.

[97] C. J. Van Rijsbergen, "Foundation of evaluation," *Journal of Documentation*, 1974.

[98] Y. Nan, K. M. Chai, W. S. Lee, and H. L. Chieu, "Optimizing f-measure: A tale of two approaches," *arXiv preprint arXiv:1206.4625*, 2012.

[99] R. Busa-Fekete, B. Szörényi, K. Dembczynski, and E. Hüllermeier, "Online F-measure optimization," in *Advances in Neural Information Processing Systems*, 2015, pp. 595–603.

[100] M. Jansche, "Maximum expected F-measure training of logistic regression models," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2005, pp. 692–699.

[101] W. Waegeman, K. Dembczyński, A. Jachnik, W. Cheng, and E. Hüllermeier, "On the Bayes-optimality of F-measure maximizers," *Journal of Machine Learning Research*, vol. 15, pp. 3333–3388, 2014.

[102] S. Decubber, T. Mortier, K. Dembczyński, and W. Waegeman, "Deep F-measure maximization in multi-label classification: A comparative study," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 290–305.

[103] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hullermeier, "Extreme F-measure maximization using sparse probability estimates," in *International Conference on Machine Learning*, 2016, pp. 1435–1444.

[104] L. Sari, M. Hasegawa-Johnson, and S. Thomas, "Auxiliary networks for joint speaker adaptation and speaker change detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 324–333, 2020.

[105] L. Sarı, S. Thomas, M. Hasegawa-Johnson, and M. Picheny, "Pre-training of speaker embeddings for low-latency speaker change detection in broadcast news," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6286–6290.

[106] "1996 English Broadcast News Speech (HUB4)," https://catalog.ldc.upenn.edu/LDC97S44.

[107] "1997 English Broadcast News Speech (HUB4)," https://catalog.ldc.upenn.edu/LDC98S71.

[108] "Switchboard-1 Release 2," https://catalog.ldc.upenn.edu/LDC97S62.

[109] "2000 HUB5 English Evaluation Speech," https://catalog.ldc.upenn.edu/LDC2002S09.

[110] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International conference for learning representations*, 2015.

[111] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[112] J. Fiscus, "NIST SCTK Toolkit," 2018. [Online]. Available: https://github.com/usnistgov/SCTK

[113] 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," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[114] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and J. Khudanpur, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," *Proc. ISCA Interspeech*, pp. 2808–2812, 2018.

[115] "Callhome Diarization Xvector Model," https://kaldi-asr.org/models/m6.

[116] S. R. Pfohl, T. Duan, D. Y. Ding, and N. H. Shah, "Counterfactual reasoning for fair clinical risk prediction," in *Machine Learning for Healthcare Conference*, 2019, pp. 325–358.

[117] D. Povey and G. Saon, "Feature and model space speaker adaptation with full covariance gaussians," in *Ninth International Conference on Spoken Language Processing*, 2006.

[118] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 5508–5518.

[119] T. Kendall and C. Farrington, "The corpus of regional African American language," *The Online Resources for African American Language Project*, vol. Version 2020.05, 2020. [Online]. Available: https://oraal.uoregon.edu/coraal

[120] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen et al., "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

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

[122] W. Labov, "The intersection of sex and social class in the course of linguistic change," *Language Variation and Change*, vol. 2, no. 2, pp. 205–254, 1990.

[123] L. Sarı, S. Thomas, and M. Hasegawa-Johnson, "Training spoken language understanding systems with non-parallel speech and text," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8109–8113.

[124] G. Adda, S. Stüker, M. Adda-Decker, O. Ambouroue, L. Besacier, D. Blachon, H. Bonneau-Maynard, P. Godard, F. Hamlaoui, D. Idiatov et al., "Breaking the unwritten language barrier: The bulb project," *Procedia Computer Science*, vol. 81, pp. 8–14, 2016.

[125] O. Scharenborg, F. Ciannella, S. Palaskar, A. Black, F. Metze, L. Ondel, and M. Hasegawa-Johnson, "Building an ASR system for a low-resource language through the adaptation of a high-resource language ASR system: Preliminary results," *Proceedings of ICNLSSP, Casablanca, Morocco*, 2017.

[126] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[127] D. Jurafsky, E. Shriberg, and D. Biasca, "Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13," University of Colorado, Boulder Institute of Cognitive Science, Boulder, CO, Tech. Rep. 97-02, 1997.

[128] E. Shriberg, R. Bates, P. Taylor, A. Stolcke, D. Jurafsky, K. Ries, N. Coccaro, R. Martin, M. Meteer, and C. Van Ess-Dykema, "Can prosody aid the automatic classification of dialog acts in conversational speech?" *Language and Speech*, vol. 41, no. 3–4, pp. 439–487, 1998.

[129] V. Raheja and J. Tetreault, "Dialogue act classification with context-aware self-attention," *arXiv preprint arXiv:1904.02594*, 2019.

[130] K. Audhkhasi, G. Saon, Z. Tüske, B. Kingsbury, and M. Picheny, "Forget a bit to learn better: Soft forgetting for CTC-based automatic speech recognition," *Proc. ISCA Interspeech*, pp. 2618–2622, 2019.

[131] L. Sarı and M. Hasegawa-Johnson, "Deep F-measure maximization for end-to-end speech understanding," *Proc. ISCA Interspeech*, pp. 1580–1584, 2020.

[132] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias," *ProPublica, May*, vol. 23, p. 2016, 2016.

[133] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS spoken language systems pilot corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[134] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[135] R. Kohavi, "Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid." in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 96, 1996, pp. 202–207.

[136] M. Redmond and A. Baveja, "A data-driven software tool for enabling cooperative information sharing among police departments," *European Journal of Operational Research*, vol. 141, no. 3, pp. 660–678, 2002.

[137] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[138] M. Kearns, S. Neel, A. Roth, and Z. S. Wu, "Preventing fairness gerrymandering: Auditing and learning for subgroup fairness," *arXiv preprint arXiv:1711.05144*, 2017.

[139] A. Cotter, M. Gupta, H. Jiang, N. Srebro, K. Sridharan, S. Wang, B. Woodworth, and S. You, "Training well-generalizing classifiers for fairness metrics and other data-dependent constraints," in *International Conference on Machine Learning*, 2019, pp. 1397–1405.

[140] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *ArXiv*, vol. abs/1405.0312, 2014.

[141] J. S. Adams, "Inequity in social exchange," in *Advances in Experimental Social Psychology*. Elsevier, 1965, vol. 2, pp. 267–299.

[142] K. Van den Bos, M. Maas, I. E. Waldring, and G. R. Semin, "Toward understanding the psychology of reactions to perceived fairness: The role of affect intensity," *Social Justice Research*, vol. 16, no. 2, pp. 151–168, 2003.

[143] G. Mikula, K. R. Scherer, and U. Athenstaedt, "The role of injustice in the elicitation of differential emotional reactions," *Personality and social psychology bulletin*, vol. 24, no. 7, pp. 769–783, 1998.

[144] L. Sarı, K. Singh, J. Zhou, L. Torresani, N. Singhal, and Y. Saraf, "A multi-view approach to audio-visual speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE (accepted), 2021.

[145] J. Stilgoe, R. Owen, and P. Macnaghten, "Developing a framework for responsible innovation," *Research Policy*, vol. 42, no. 9, pp. 1568–1580, 2013.

[146] "Digital responsibility website," [Last access: Jan 29, 2021] http://www.digitalresponsibility.org/health-and-technology.

[147] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *Proc. Interspeech 2019*, pp. 161–165, 2019.

[148] S. Rongali, B. Liu, L. Cai, K. Arkoudas, C. Su, and W. Hamza, "Exploring transfer learning for end-to-end spoken language understanding," *arXiv preprint arXiv:2012.08549*, 2020.