AUTOMATIC SPEECH RECOGNITION SYSTEM ADAPTATION FOR SPOKEN LECTURE PROCESSING

by

Enver Fakhan

B.S., Electrical and Electronics Engineering, Mersin University, 2016

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2021

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my co-advisor Assist. Prof. Ebru Arısoy for her guidance, support and most importantly patience. This work absolutely couldn't be finished without her support and her believing in me. I haven't been the easiest student but she has been the best advisor. I feel incredibly lucky that I happened to know her in this chaotic world. Her kindness, her support, her meticulous way of working will stand as a prominent example throughout my life. I would also like to present my gratitude to my co-advisor Prof Murat Saraçlar for his critical feedbacks and advises during this work and special thanks for his invaluable classes. I would like to mention Prof. Levent Arslan and Assoc. Prof. Cenk Demiroğlu for accepting to be in my defense jury and sharing their invaluable feedback.

I'm grateful to the friends and colleagues at BUSIM lab. namely Öykü Deniz Köse, Alican Gök, Can Gürsoy, Gözde Çetinkaya, Bolaji Yusuf, Nazif Can Tamer, Korhan Polat for their welcoming receive in my early days at the lab and their precious feedbacks during discussions. I would like to give a special thank to Nazif Can Tamer for the interesting discussions we have in such broad ranges.

I can't express my gratitude enough to Korhan Polat for his support, feedbacks, advises and insights during the whole time. He was always there when I needed a sane advise in the darkest times.

I would also like to thank my friends for their interest, namely Sorguç Heval Rengis, Birsu Çekin, Güzel Durmaz, Can Uysal, Mehmet Demirtaş, Özgür Yılmaz and Abdurrahman Gügercin.

I would like to thank my family for believing in me. Life has not been in the easiest times but I guess we are working on it. Last but not least, I would like to dedicate this work to my newly born niece Arin, I hope she finds a liveable world by

the time she is grown up, and furthermore she has an exciting and joyful life.

I would like to thank the faculty members of MEF University for giving permission for using their flipped learning videos for this research.

This work was supported by the TÜBİTAK-ARDEB 3501 Program (Project No: 117E202).

ABSTRACT

AUTOMATIC SPEECH RECOGNITION SYSTEM ADAPTATION FOR SPOKEN LECTURE PROCESSING

The recent developments in artificial neural networks has brought significant improvement in Automatic Speech Recognition (ASR). However the performance of the neural network based models mostly depends on the availability of large amounts of data and computational resources. When there is limited amount of in-domain data, acoustic and language model adaptation methods are used. These methods utilise large amount of out-of-domain data as well as limited in-domain data while learning the parameters of the model. This work explores different adaptation methods in neural network based ASR systems developed for spoken lecture processing in English and in Turkish. We mainly investigate speaker adaptation, acoustic condition adaptation and effect of both adaptations together with limited amount of spoken lecture data. We show that building a source model with out-of-domain data and adapting this model with limited in-domain data yields improvement in performance both in hybrid acoustic model based ASR systems and in end-to-end ASR systems.

ÖZET

SÖZLÜ DERS ANLATIMLARININ İŞLENMESİ İÇİN OTOMATİK KONUŞMA TANIMA SİSTEMİNİN UYARLANMASI

Yapay sinir ağlarındaki son gelişmeler, Otomatik Konuşma Tanıma'da (OKT) önemli iyileştirmeler getirmiştir. Bununla birlikte, sinir ağı tabanlı modellerin performansı çoğunlukla büyük miktarda verinin ve hesaplama kaynaklarının mevcudiyetine bağlıdır. Sınırlı miktarda alan içi veri olduğunda, akustik ve dil modeli uyarlama yöntemleri kullanılır. Bu yöntemler, modelin parametrelerini öğrenirken büyük miktarda alan dışı veri ile birlikte sınırlı alan içi veri de kullanır. Bu çalışma, İngilizce ve Türkçe sözlü ders anlatımlarını işleme için geliştirilen sinir ağı tabanlı ASR sistemlerinde farklı uyarlama yöntemlerini araştırmaktadır. Biz temel olarak konuşmacı uyarlama, akustik durum uyarlama ve her iki uyarlamanın birlikte yapılmasının etkisini sınırlı miktarda sözlü ders anlatımları verisi ile araştırıyoruz. Alan dışı veri ile bir kaynak model oluşturmanın ve bu modeli sınırlı miktarda alan içi veri ile uyarlamanın hem hibrit akustik model tabanlı OKT sistemlerinde hem de uçtan uca eğitilen OKT sistemlerinde başarım artışı sağladığını gösteriyoruz.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_t	The Sub-word at time t in a given alignment A for CTC training
$E[\cdot]$	Expectation operation
L_{CE}	Cross entropy loss function
$M(W_u)$	HMM state sequence for word sequence belonging to utterance u
o_τ	Acoustic input at time step τ
O_u	Acoustic sequence belonging to utterance u
$std(\cdot)$	Standard deviation of the input
U	Utterance set in the training data
W_u	Word sequence of the utterance u
$\hat{W}$	Predicted word sequence by the decoder
x_τ	Input feature vector at time step τ
z_τ	Hidden representation of the input vector x_τ generated by the acoustic encoder
$\mathcal{A}_{\mathcal{X},\mathcal{Y}}$	Set of valid alignments for an utterance in CTC training
τ	Time Step in the acoustic input sequence

LIST OF ACRONYMS/ABBREVIATIONS

ADAM	Adaptive Moment Estimation
AM	Acoustic Model
ASR	Automatic Speec Recognition
BPE	Byte Pair Encoding
CD	Context-Dependent
CE	Cross Entropy
CI	Context-Independent
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
DNN	Deep Neural Network
fMLLR	Feature Space Maximum Likelihood Linear Regression
GMM	Gaussion Miture Model
HMM	Hidden Markov Model
LDA	Linear Discriminant Analysis
LF-MMI	Lattice Free-Maximum Mutual Information
LM	Language Model
LSTM	Long Short-Term Memory
MFCC	Mel-Frequency Cepstrum Coefficients
MHA	Multi-Head Attention
MLLT	Maximum Likelihood Linear Transform
MMI	Maximum Mutual Information
NLP	Natural Language Processing
NN	Neural Network
RNN	Recurrent Neural Network
TDNN	Time Delay Neural Network
WER	Word Error Rate

1. INTRODUCTION

The task of Automatic Speech Recognition (ASR) is to transduce an acoustic sequence into its textual representation. Historically Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) based acoustic models have been used in ASR systems until the recent developments in neural network based modeling. While the neural network based ASR systems showed successful results [3–6], they usually require large amounts of data and computational resources for training. Furthermore, the performance of ASR systems are mostly data set dependent [7], that is, the performance of an ASR system trained on a particular data set usually doesn't translate into other data sets. Because of these reasons, either an ASR system is developed with the in-domain data or adaptation methods are used when there is limited amount of data to develop an ASR system from scratch.

The contemporary adaptation approaches in ASR ranges from unsupervised pre-training methods [8] to transfer learning [9] and multi-task learning [10, 11] with supervised methods. In the multi-task learning, usually parameters in some parts of the model is shared among different tasks and some layers are reserved to each task during training.

In transfer learning with supervised methods, first a source model is trained using the generic data set and then the whole model or some part of the model is adapted to a different but related task or domain [9, 12]. Since the training of the source model is done with supervised learning, this approach requires much less data compared to the pre-training approach. However the benefit of the transfer learning decreases when the dissimilarity between the source and the target datasets increases [13].

To utilise adaptation in transfer learning as much as possible, the dataset for the source model should be picked carefully. A discrepancy in gender and dialects among the speakers can cause a bias in the source model [14] and then this can transfer into

the target model. Since improvements in one speaking style do not always transfer to other speaking styles [15], it is crucial that the source data set and the target data set have similar speaking styles (planned vs spontaneous). Another important factor is the variation in the acoustic condition. A mismatch in the acoustic condition between the source and the target datasets can prevent doing a successful adaptation.

In this thesis, we build various ASR systems for spoken lecture processing. We use generic datasets that have similar characteristics with in-domain datasets to train a source model. Then the source model is adapted with in-domain datasets. We investigate the adaptation methods both for ASR systems with hybrid acoustic models and end-to-end ASR systems. In order to assess the efficacy of the adaptation methods, we also build baseline models with available in-domain data. We investigate the effect of speaker adaptation and acoustic condition adaption separately by dividing in-domain data into different sets.

Online learning has been an increasing practice in recent decade in many institutions. The Covid-19 pandemic has accelerated this process even further with many institutions going completely remote-teaching in this period. Having an ASR system for spoken lecture processing can assist learner significantly not only by providing transcriptions for the video lectures but also with its usage in down stream tasks such as Keyword Search (KWS).

The main contributions of this thesis are as follows;

(i) We collected spoken lecture data in English and in Turkish.
(ii) We built hybrid acoustic model and end-to-end ASR systems for spoken lecture processing.
(iii) We utilised large amounts of out-of-domain data for building systems and applied adaptation with in-domain data using transfer learning to remedy the limited data in the lecture domain.
(iv) We investigated both speaker adaptation and acoustic condition adaptation.

(v) We compared the performance of hybrid acoustic model and end-to-end ASR systems in adaptation.

The rest of the thesis is organised as follows:

- In Chapter 2, background knowledge on ASR systems is provided.
- In Chapter 3, the adaptation approaches used in our research are explained.
- In Chapter 4, the dataset used for training and adaptation is described.
- In Chapter 5, the empirical results and discussion are provided.
- In Chapter 6, conclusion is given.

2. BACKGROUND

This chapter provides background information both on Neural Network-Hidden Markov Model (NN-HMM) and end-to-end ASR systems.

2.1. Statistical ASR

The task of Automatic Speech Recognition is to transduce a sequence of acoustic features that belong to an utterance into its linguistic representation. This can be expressed as predicting the most likely word sequence, $\hat{W}$, given the acoustic features, X. This is formularised as:

$$\begin{aligned} \hat{W} &= \arg\max_{W} P(W \mid X) \\ &= \arg\max_{W} P(X \mid W) P(W) \end{aligned} \tag{2.1}$$

where W is a word sequence. Here $P(X \mid W)$ is obtained with the acoustic model (AM) and $P(W)$ is obtained with the language model (LM). For the acoustic model, words are generally split into smaller sub-units like sub-words, phonemes, or context dependent phonemes. The acoustic model assigns a probability to each acoustic feature vector given a sub-unit and the language model gives the probability to each sequence of words. The language model reduces the search space and enables to distinguish acoustically similar word sequences by assigning higher probability to semantically more likely word sequences [16]

2.2. GMM-HMM Acoustic Models

Before the resurgence of neural networks, the GMM-HMM acoustic modelling was state of the art for ASR. The GMM is used to model the likelihood of an acoustic input given the HMM state, and the HMM is used to model temporal variation in speech. Likelihood of states are combined with other knowledge sources like lexicon

and language model to construct a search graph. Then via a decoder possible text sequence hypothesises for acoustic sequence are extracted from search graph. Training is performed with Expectation Maximisation algorithm.

A typical GMM-HMM training consist of multiple stage training. In the first stage, words are represented with Context Independent (CI) phonemes and GMMs are initialised with flat start (i.e phonemes are aligned to feature vectors with equal amount of time). After CI phoneme based GMM-HMM converges, training set is aligned with this model. Then in the second stage, CI phonemes are replaced with Context Dependent (CD) phonemes (typically tri-phone). Using previously aligned training dataset, a new CD GMM-HMM is initialised and trained until it converges. Finally this model is refined by doing an alignment followed by a retraining procedure with additional transformation applied to input vector like Linear Discriminant Analysis (LDA) Maximum Likelihood Linear Transform (MLLT) and Feature Space Maximum Likelihood Linear Regression (fMLLR) [17, 18].

2.3. NN-HMM Hybrid Models

In hybrid acoustic models [19], a neural network is used to produce posterior distributions over tied HMM states for each frame. These posteriors are divided by states' prior distribution to obtain likelihood of acoustic input given the HMM state. Then the likelihood of the states are integrated into search graph for decoding.

Although hybrid systems are less appealing compare to end-to-end neural transducer approaches, they have the advantages of easily integrating other knowledge sources, such as specialised lexicons [20]. They also require less data for training a neural network model compared to training a neural network model in an end-to-end setting that is comparable in size [21].

2.3.1. Cross Entropy Training

In the Cross Entropy (CE) training, the neural network acoustic model is trained with the objective of maximising the log-likelihood of a given HMM state alignment for an acoustic feature sequence [22]. Since the output of the neural-network is a posterior-like distribution, maximising the the log-likelihood of the given HMM state is achieved by minimising negative cross entropy loss between the posterior distribution produced by the acoustic encoder and the given state alignment. Then the loss function for the neural-network becomes;

$$L_{CE} = -\sum_{u \in U} \sum_{\tau=0}^{T_u} \log P(M_\tau(W_u) \mid o_\tau) \tag{2.2}$$

where U is the utterance set, W_u is the word sequence for the utterance u and $M_\tau(W_u)$ is the HMM state at time τ in the corresponding HMM state alignment and, the o_τ is the acoustic input at time τ. The HMM state alignment for the training set is obtained through a previously trained GMM-HMM model.

During inference the posterior distribution produced by the encoder is divided by the state priors to obtain likelihood of the states for decoding.

2.3.2. Maximum Mutual Information Training

In the MMI training, the acoustic encoder is trained by maximising mutual information between acoustic sequence and the corresponding word sequence [23]. The objective function F_{MMI} is;

$$F_{MMI}(\theta) = \sum_{u \in U} \log \frac{P_\theta(O_u \mid M(W_u))P(M(W_u))}{\sum_{\hat{u} \in U} P_\theta(O_u \mid M(W_{\hat{u}}))P(M(W_{\hat{u}}))} \tag{2.3}$$

where O_u is the acoustic sequence for the utterance u and $M(W_u)$ is the corresponding HMM state sequence. The numerator is the likelihood of the acoustic sequence

given the correct word sequence, and the denominator is the total likelihood of the acoustic sequence given all possible word sequences in the dataset. The computation of the denominator requires to loop through all the possible word sequences and it is extremely expensive. To overcome this issue, two methods has been proposed, these are lattice-based and lattice-free methods. Their details are explained in the following sub-sections.

In the lattice based approach, instead of computing the total likelihood of an utterance with all possible given word sequences, the total likelihood is approximated with a lattice containing n-best hypothesises obtained through decoding the utterances with a previously trained GMM-HMM system [23].

In the lattice-free approach, the corresponding HMM sequence of the utterances is generated with a phone level sequence instead of a word sequence [24]. Additionally, to reduce the total number of states even further a decoding graph is generated by composing all HMM sequences. Finally the total likelihood of the utterance is computed with this graph.

2.3.3. Neural Network Architectures in Hybrid Models

Different neural network architectures can be used to obtain posterior distribution of HMM states. In Deep Neural Network (DNN), Time Delay Neural Network (TDNN) and Convolutional Neural Network (CNN) based architectures, the posterior distribution of HMM states for a frame x_τ is conditioned on its neighbours. The width of the context in DNN based architectures is designated by how many acoustic features are concatenated, in CNN based architectures it is a function of kernel sizes, strides and the depth of the network architecture. In uni-directional RNN architectures, the posterior distribution of HMM states for acoustic frame x_τ is conditioned on the whole history, $x_0, x_1, ..., x_\tau$, while in bi-directional RNN architectures it is conditioned on the entire input sequence. In the recently proposed transformer based architectures, the posterior distribution is also based on its entire input sequence, while this can be

adjusted by a masking procedure.

The details of the architectures used in the experiments with hybrid acoustic models are explained in the following sub-sections.

2.3.3.1. DNN-CE. Before training with DNN-CE model, first a GMM-HMM model is trained to obtain HMM state alignments for the training data. The number of tied HMM states in the GMM-HMM model determines the number of output classes in the DNN-CE acoustic model.

The DNN-CE acoustic model has a DNN based network architecture and is trained with cross entropy objective function. In the experiments, the p-norm non-linearity function is used in-between layers for this architecture. The p-norm non-linearity function is $y = \|\mathbf{x}\| = (\sum_i |x_i|^p)^{1/p}$ where p is 2 and the vector $\mathbf{x}$ represents a group of inputs [25]. The p-norm non-linearity function reduces the dimension of the hidden vectors based on the group size. This also reduces the total number of parameters in the model [25].

The acoustic input vectors are concatenated with a fixed number of context in both directions, $\hat{x}_i^T = [x_{i-l}^T, x_{i-l-1}^T, ..., x_i^T, ..., x_{i+l-1}^T, x_{i+l}^T]$. A transformation matrix is constructed with Linear Discriminant Analysis (LDA) using aligned training data, and the input vector $\hat{x}_i$ is pre-conditioned before being fed to the network.

At the last layer, softmax function is applied to approximate posterior distribution of HMM states. The network is trained by minimising the negative cross entropy between produced posterior distribution and the aligned HMM state, as explained in the section 2.3.1

2.3.3.2. TDNN-CE. Time Delay Neural Network (TDNN) uses time dilation in hidden layers to represent relationship between events in time [26]. Contrary to DNN based architectures, where all the layers in the network learns representations for the same

context, in TDNN based architectures, the first layer learns representations for a narrow context while the higher layers learns representations for a wider context. Hidden vectors at each layer are sub-sampled, which results in reduction in computation and model size [6]. The forward pass procedure with sub-sampling is shown in the Figure 2.1

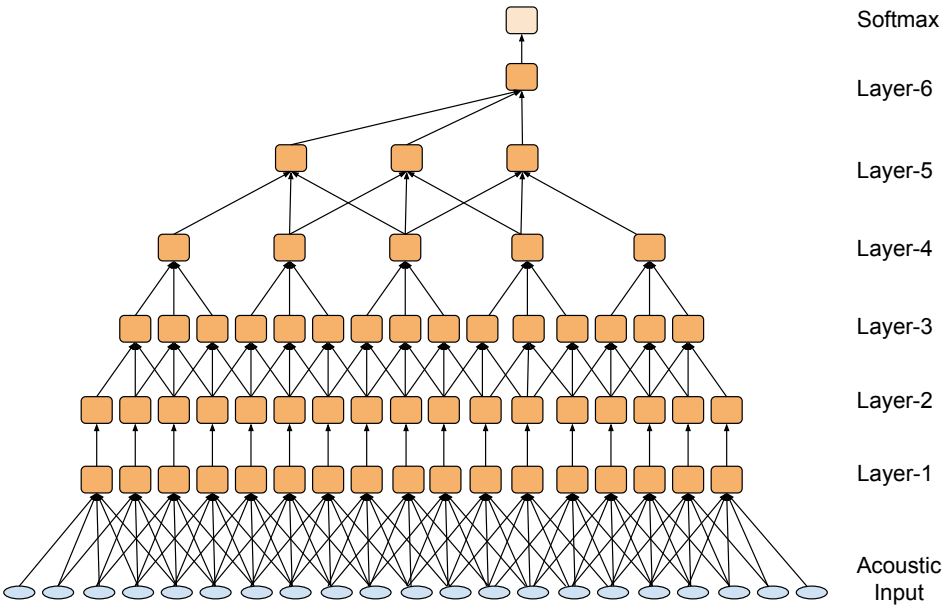

Figure 2.1. Computation in TDNN with different input context at each layer

As with the DNN-CE model, a previously trained GMM-HMM model is used to obtain HMM state alignments for the training data. In the experiments, the training set is augmented with speed and volume perturbation [6]. The input features are concatenation of 5 40 dimensional Mel-Frequency Cepstrum Coefficients (MFCC) and a 100 dimensional i-vector representation, which, in total, is a 300 dimensional vector. The model is trained with cross entropy loss function as in the DNN-CE training.

2.3.3.3. TDNN-LF-MMI. TDNN-LF-MMI acoustic model has a similar network architecture with the TDNN-CE acoustic model, however it is trained with Lattice-Free Maximum Mutual Information (LF-MMI) objective function. Training a TDNN-LF-

MMI model doesn't require a prior GMM-HMM training to obtain HMM state alignment for acoustic sequences. However a previously trained GMM-HMM model is used to obtain phone-level alignments of the training data because the forward-backward computation for denominator graph is memory intensive [24]. These alignments are used for building a phone level n-gram language model to construct a denominator graph. Additionally, the phone-level alignments of the training data are utilised to split utterances into less than 1.5 seconds of chunks. And lastly, to reduce the computation cost further, the LF-MMI objective function is computed using neural network outputs at one third of the standard frame rate [24]. Dependency to a previously trained GMM-HMM model can be avoided with adequate computation resources and some modification to HMM topology [27]. In addition to LF-MMI training, the TDNN-LF-MMI model is regularised with cross entropy loss function using soft alignments obtained in forward-backward computation.

2.3.4. Language Model in Hybrid Systems

Language models in the hybrids are usually n-gram based models which can naturally be integrated into the search graph. However, neural-network based language models may also be used for re-scoring the n-best hypothesis or the lattices obtained through a decoder [28].

N-gram language models are statistical models that assign a probability to the next word given the previous $n-1$ words. The n-gram language models are trained with large text corpora, however some of the n-grams may not be present in the text corpora. To overcome this problem different smoothing and/or back-off methods are applied [29, 30].

2.4. End-to-End Neural Transducer

In end-to-end neural transducer systems [31–33], an acoustic encoder is used to embed acoustic input sequence, $x_1, x_2, ...x_T$, into its high level representations,

$z_1, z_2, ..., z_T$, and a neural network decoder is used to obtain sequence hypotheses at the grapheme level conditioned on the embedded representations. The encoder and the decoder are jointly trained.

2.4.1. CTC Training

Connectionist Temporal Classification (CTC) is a way of assigning a probability to a mapping of an input sequence $X = [x_1, x_2, ...x_T]$, such as audio, to an output sequence $Y = [y_1, y_2, ...y_U]$, such as transcription, when there is no alignment present between them [34]. To obtain probability of a sequence, CTC marginalises over all valid alignments for the output sequence Y. More precisely, for a given single pair (X, Y), the CTC objective is to maximise;

$$P(Y \mid X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^{T} p_t(a_t \mid X) \tag{2.4}$$

CTC collapses repeated characters in an alignment to obtain its corresponding output sequence. However a straightforward collapsing would fail to capture an output sequence which contains repeated characters in it, such as *hello*. To overcome this problem, CTC introduces a new token ϵ. This token is placed at the beginning, end, and between every token in the output sequence. By doing so, the target sequence becomes $Z = [\epsilon, y_1, \epsilon, y_2, \epsilon, ...\epsilon, y_U, \epsilon]$. After repeated characters are collapsed, finally ϵ token is removed and an alignment for the corresponding sequence is obtained. An example of this process is shown in Figure 2.2. Hence every alignments that collapses into the target sequence after this process is a valid alignment.

Marginalising over all valid alignments can be too slow to compute. As a work around, dynamic programming is used to compute the sum of valid alignment probabilities. To achieve this, each token in the target sequence Z, is allowed to have transitions to itself and to the next token. To account for the alignments where ϵ symbol doesn't occur between two tokens, each token that is present in the original sequence Y is allowed to have a transition to the next token in the sequence Y, unless they are the

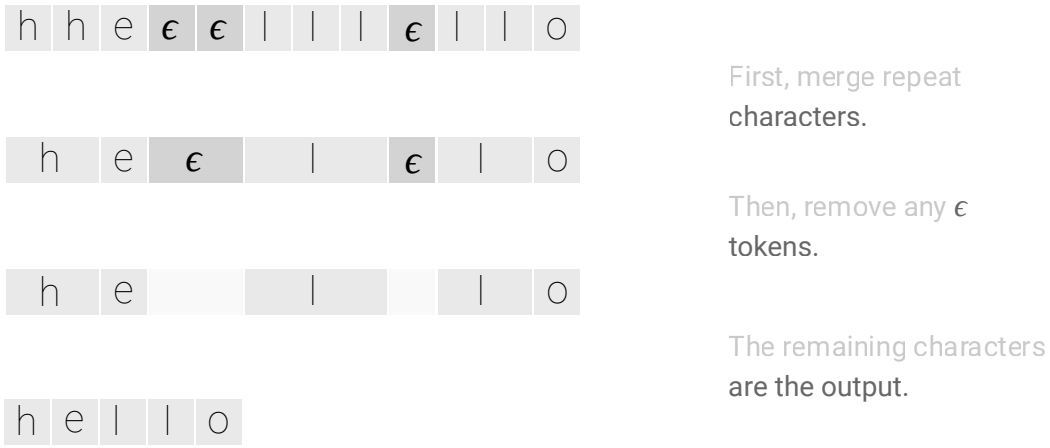

Figure 2.2. CTC alignment steps. This figure is taken from [1] and it is licensed under Creative Commons Attribution CC-BY 4.0 [2].

same token. An example of this computation for an acoustic input sequence of length 6 and output sequence of $[a, b]$ is shown in Figure 2.3.

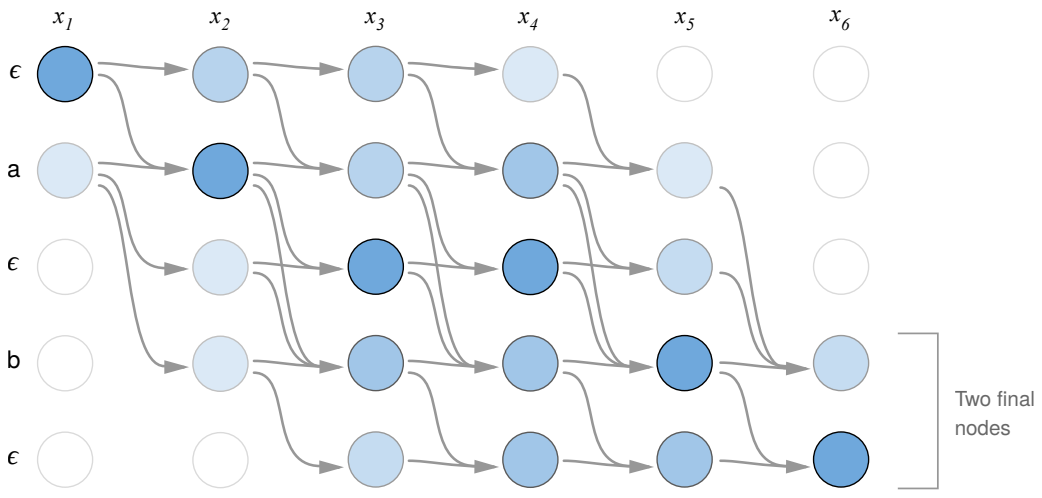

Figure 2.3. CTC cost computation with dynamic programming. This figure is taken from [1] and it is licensed under Creative Commons Attribution CC-BY 4.0 [2].

2.4.2. Byte Pair Encoding

Byte Pair Encoding (BPE) is used to split words into its sub-tokens to create targets for the end-to-end model training. BPE originally proposed as a data compression algorithm [35] that iteratively replaces the most frequent pair of bytes in a sequence with a single unused byte. BPE is slightly modified to perform sub-word tokenization,

such that, instead of replacing a pair of tokens with the new one, they are merged and added to the token set.

The algorithm works as follows: first the symbol vocabulary is initialised with single characters and words are represented with the current symbol set and a special character that indicates the end of word, </w>. Then, the algorithm loops through all the words and counts the occurrences of consecutive tokens. Finally the most frequent pair of tokens is merged in all occurrences and added to the vocabulary of symbols. This process is repeated until the number of symbols in the vocabulary equals the desired vocabulary size. BPE enables even the encoding of the unseen words by splitting them into known tokens that is present in the vocabulary. An example for this could be the following; *athazagoraphobia = ['_ath', 'az', 'agor', 'aphobia']*. BPE would only introduce unknown token when there is a symbol in the word that is not present in the token set.

2.4.3. Acoustic Model in End-to-End Systems

Different neural network architectures can be used to model the encoder and the decoder in the end-to-end ASR systems.

The speech transformer [36] acoustic model is a transformer [37] based encoder-decoder network inspired by its success in the NLP field [38, 39].

The encoder part is composed of encoder-transformer blocks and a pre-processor module, M, which consists of convolutional layers. An encoder-transformer block consists of a Multi Head Attention (MHA) layer followed by a two-layer feed forward neural network. Layer norm [40] is applied to hidden vectors both before the MHA layer and the feed forward network. There are two residual connections in a encoder-transformer block, one adds the inputs (before layer norm is applied) to MHA layer to its output, and the other one adds the input (before layer norm is applied) to feed forward neural network to its output. The acoustic input sequence is pre-processed and down-sampled

by the pre-processor module M before being passed to the encoder-transformer.

The decoder part consists of only decoder-transformer blocks. A decoder-transformer block is similar to an encoder-transformer block, except it has an additional masked MHA layer preceding to the MHA layer. The masked MHA layer is used to attend only the previous tokens, and the regular MHA layer is used to attend the contextual acoustic representations. A depiction of speech transformer is shown in Figure 2.4

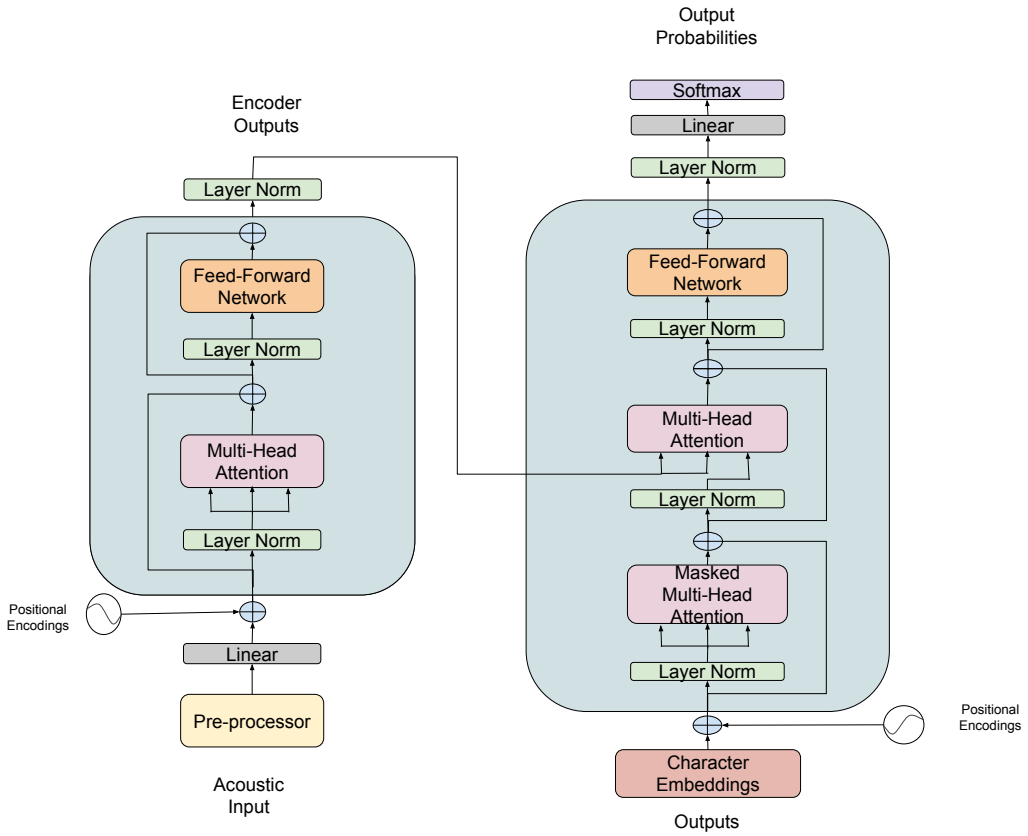

Figure 2.4. Speech Transformer

2.4.4. Language Model in End-to-End Systems

Language models in End-to-End systems are usually neural-network based models which can be used for rescoring n-best hypothesis or be integrated to the system during decoding [33]. Different neural-network architectures can be used to train a language model.

3. ADAPTATION METHODS

In neural networks, multi-task learning has been employed to learn intermediate level features that are useful for various tasks [10, 11]. Pretraining is another possible approach to implicitly learn intermediate representations for different tasks [41, 42]. It has been shown that the intermediate layers in neural networks trained on speech data discover useful representations for various tasks, while the higher layers learn task specific representations [43]. Multi-task learning in speech processing can be applied to the same task (e.g. speech recognition) but on different domains (conversation vs read speech) or it can be applied to different tasks (phoneme recognition vs audio classification).

Numerous transfer learning approaches have been used in speech recognition for adaptation purposes [9]. In the early works, domain and speaker adaptation have been attempted with adapting network parameters using Linear Input Network [44]. This has lead to more advanced methods like Linear Hidden Networks (LHN) in which the adaptation is employed to a newly added linear network that is placed between the last hidden layer and the output layer [45]. It has been shown that multi-task learning by training on different level of phonemes (monophone, senone) supported by LHN adaptation helps to deal with unseen senone problem [46]. Several transfer learning approaches, including multi task learning, has been studied for speaker adaptation [47].

Recently popularised weight transfer methods can be taught as a type of LHN based adaptation. In the weight transfer approach the last layer of the network is usually not transferred for the reason that the phone set in the source domain and in the target domain are different. Hence the last layer is either replaced with a domain specific layer or a new layer is added and the whole or a part of the network is adapted [12]. However, if the source and the target domain share the same phone set, the last layer modification may not be needed. In fact it is shown that the best improvement in Word Error Rate (WER) is obtained when the adaptation was done

on the whole network with the last layer included while adapting a model trained on Librispeech data with WSJ data [12]. This is particularly useful when the target dataset is very small compared to the source dataset [48].

In this work we apply adaptation with transfer learning in 2 different setups for spoken lecture domain adaptation in both hybrid acoustic models and the acoustic models in the end-to-end systems. To apply transfer learning for cross entropy based hybrid models, first frame level alignments of the adaptation sets are obtained using the GMM-HMM source model trained with the out-of-domain data. Then, the cross-entropy based hybrid source model is adapted using frame level alignments of the adaptation data. For the LF-MMI based hybrid models, lattices that contain different pronunciations for the adaptation dataset are generated first using the same GMM-HMM source model. Then using the lattices, the nominator and denominator graphs are built for adaptation sets that will be used for transfer learning on LF-MMI based hybrid source models. In the end-to-end system, tokenizing the transcription of the adaptation set with the same tokenizer used in the source model is sufficient to apply transfer learning with CTC training. The details of the adaptation setups are described in the following sections.

3.1. Adaptation in Setup-1

In this approach we train the source model with an out-of-domain data and then adapted the whole network with an in-domain adaptation data. A depiction of this process is shown in Figure 3.1. In order to adapt the whole network without modifying the output layer, we use a phone set for the source models that encompasses the phone set for the target data in the hybrid acoustic models. For the acoustic model in the end-to-end, simply using the same tokenizer with the same vocabulary enables us to adapt the whole network. For the hybrid-acoustic models, we simply pick the last model obtained during source model training and we retrain it with the adaptation data for a certain number of epochs. For the end-to-end models, we train the source model for a fixed number of epochs and then we pick a model at a recent epoch for

adaptation. We retrain the model with the adaptation data with an early stopping criteria.

For the purpose of assessing the contribution of speaker adaptation and acoustic condition adaptation separately, we employ adaptations in three different settings. In the first setting we use single speaker data from target dataset for adaptation. The development and the evaluation datasets contain speeches from only this speaker, hence we would be able to asses the contribution from speaker adaptation. In the second setting we exclude this speaker and adapt the source model with the remaining data. Doing so, we would be able to asses acoustic condition adaptation free from speaker adaptation effect. In the last setting we use all target data available for adaptation, this is considered as acoustic condition and speaker adaptation together.

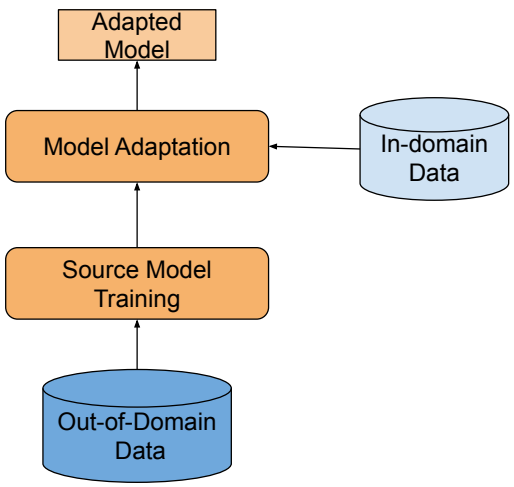

Figure 3.1. Adaptation method in Setup-1.

3.2. Adaptation in Setup-2

To mimic multi-task learning with multi-domain training, we mix the target dataset with source dataset while building the source model. A depiction of this process is shown in Figure 3.2. Because we use the same phone set in hybrid acoustic

model for both source data and target data we do not use separate last layers for the source domain and the target domain. Instead we explicitly share all the parameters by doing training on the mixed dataset. Since we use all the adaptation dataset during training the source model, we do not apply speaker adaptation and acoustic condition adaptation analysis in this setup as we did in Setup-1. We apply the same transfer learning approach to the source model as we did in Setup-1 approach, which is, we fine-tune the hybrid models with adaptation set for a fixed number of epochs.

Even tough the adaptation in Setup-2 has the advantage of getting a better source model by exposing it to the adaptation set during training, it needs to be trained from scratch every time a new batch of in-domain data arrives. However, the source model in Setup-1 can be trained for once and later it can be adapted every time a new batch of in-domain data arrives.

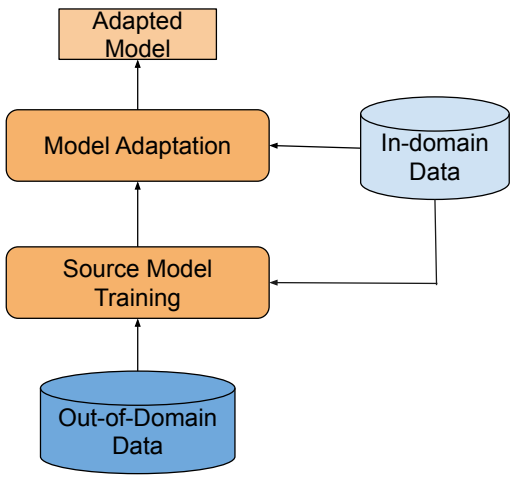

Figure 3.2. Adaptation method in Setup-2.

4. DATASET

This chapter explains English and Turkish datasets used in ASR system development and adaptation.

4.1. English Dataset

4.1.1. English Lecture Dataset

For English lecture dataset, we have collected video lectures prepared at MEF University for flipped learning. The video lectures were recorded in a studio setting and they mostly contain planned speech.

The English lecture dataset is composed of 119 video lectures, representing about 10 hours of speech, of which female speakers comprise about 7 hours and 45 minutes and male speakers comprise about 2 hours and 15 minutes. The dataset contains courses from engineering (6 hours) and social sciences (4 hours). The dataset is divided into disjoint adaptation, development and evaluation parts. The development and evaluation sets contain video lectures from one female speaker. The development set is composed of 8 video lectures for Circuit Analysis course and contains about 45 minutes of speech. The evaluation set contains 15 video lectures for Signal and Systems course and contains about 1 hour and 15 minutes of speech. The remaining data is set apart as the adaptation part.

The adaptation part of the English lecture dataset is divided into 2 parts. These parts are named as SLP-1 and SLP-2 adaptation sets. SLP-1 adaptation set contains 27 video lectures given by one female speaker with a total duration of 2 hours and 15 minutes. This speaker is the same speaker that development and evaluation sets are composed of. In other words the video lectures in SLP-1, development and evaluation sets are all prepared by the same speaker. The SLP-2 adaptation set contains 69 video

lectures for engineering and social sciences courses with a total duration of 5 hours and 45 minutes. There are 3 female and 3 male speakers in this set. The details of the English lecture dataset are summarised in Table 4.1 and 4.2[1] .

Table 4.1. Text characteristic of English lecture dataset.

	SLP-1	**SLP-2**	**Development**	**Evaluation**
Number of lectures	27	69	8	15
Number of segments	1073	5485	385	652
Number of words in transcriptions	16 735	43 064	5574	9787

Table 4.2. Audio characteristic of English lecture dataset.

Characteristics	**SLP-1**	**SLP-2**	**Development**	**Evaluation**
Total duration	2h 15m	5h 45m	45m	1h 15m
-Female	2h 15m	3h 30m	45m	1h 15m
-Male	0	2h 15m	0	0
Mean duration	5m 4s	5m	5m 44s	4m 58s
Unique speaker	1	6	1	1
-Female	1	3	1	1
-Male	0	3	0	0

To build a text corpus for English lecture dataset, we collected transcriptions of MIT open lectures given in Digital Signal Processing [49], Circuit Analysis [50] and Signal and Systems [51] courses. We also use the transcriptions of SLP-1 adaptation dataset along with content of the slides for lectures in SLP-1 set. It contains 23.215 sentence composed of a total of 400.1k words.

4.1.2. TEDLIUM

Tedlium dataset [52] is a curated corpus developed for building ASR systems, extracted from TED video talks. The recordings are generally uninterrupted and planned

[1]h: hour, m: minute, s: second.

talks as in the English lecture dataset.

Table 4.3. Tedlium corpus text characteristics.

	Train	**Dev**
Number of talks	774	19
Number of segments	56.8k	2k
Number of words	2.56M	47k

The training set of the corpus is composed of 774 talks, representing 118 hours of speech. This set consists of 666 unique speakers, of which male and female speakers constitute 82 and 36 hours of speech respectively. The development set is composed of 19 talks, representing 4 hours of speech, which contains 3 hours of male and 1 hour of female speakers. The characteristics of the corpus in terms of text and audio are shown in more detail in Table 4.3 and 4.4 respectively.

Table 4.4. Tedlium corpus audio characteristics.

	Train	**Dev**
Total duration -Male -Female	118h 4m 48s 81h 53m 7s 36h 11m 41s	4h 12m 55s 3h 13m 57s 58m 58s
Mean duration	9m 9s	13m 18s
Number of unique speakers	666	19

A text corpus is also shared with Tedlium data for language modelling. The total number of sentences in the corpora is about 12.2M, the details are given in Table 4.5. The total number of words is about 229.19M.

Table 4.5. Collection of corpus for LM training.

Corpus	**Number of Sentence**
Common Crawl	1.194.029
Europarl V7	180.541
10^9 FR-EN	900.530
News-com. v8	32.234
News	9.593.018
Yandex 1M	350.000
Total	12.250.352

4.2. Turkish Dataset

4.2.1. Turkish Lecture Dataset

For Turkish lecture dataset, we have collected video lectures prepared for Law courses offered in Turkish at MEF University for flipped learning. The video lectures were recorded in a studio setting and they mostly contain planned speech. The dataset is divided into disjoint adaptation, development and evaluation sets.

The adaptation set, which is called SLP-3, is composed of 194 video lectures coming from 17 different law courses. The total duration of the dataset is about 31 hours and 17 minutes. It contains 4 female and 4 male speakers. The mean duration of the recordings is about 9 minutes and 40 seconds and they are split into 27.488 segments with mean duration of 3.2 seconds. The data set contains 183.970 words in total.

The development set is composed of 22 video lectures coming from three different law courses with a total duration of about 4 hours. It contains 2 male speakers and 1 female speaker. Total duration of the recordings coming from the female speaker is about 2 hours and 14 minutes. While the SLP-3 adaptation dataset contains some recordings that belongs to the male speakers, it doesn't contain any recording from

Table 4.6. Audio characteristics of Turkish lecture data.

	SLP-3	**Developement**	**Evaluation**	**Total**
Number of Lectures	194	22	28	244
Total Duration	31h 17m	4h	3h 51m	39h 8m
Male	18h 14m	1h 46m	2h 59m	22h 59m
Female	13h 3m	2h 14m	0h 52m	16h 9m
Number of Unique Speaker	8	3*	4*	11

the female speaker. The mean duration of the recordings is about 10 minutes and 48 seconds and they are split into 4081 segments with a mean duration of 3.2 seconds. The development set contains 24.276 words.

The evaluation set is composed of 28 video lectures coming from 4 different law courses with a total duration of about 3 hours and 51 minutes. It contains 3 male speakers and 1 female speaker. The recordings belong to the female speaker comprise of about 52 minutes. The SLP-3 adaptation dataset contains some recording from the two of the male speakers while others are excluded. The recordings have about a mean duration of 8 minutes and 16 seconds and they are split into 3146 segments with a mean duration of about 4 seconds. The evaluation set contains 20.793 words.

The details of the dataset are shown in Table 4.6. Two speakers from the development set and 2 speakers from the evaluation set have recordings in SLP-3 adaptation set, this is indicated with an asteriks symbol in the table.

4.2.2. TuskishBN dataset

Turkish Broadcast News (TurkishBN) dataset is a collection of news segments from 4 news outlets curated for ASR applications [53]. It contains about 194 hours of speech and 1.3M words in the reference transcriptions. The dataset is partitioned into disjoint training, held-out, and test sets. Tuskish News corpus is a collection of news collected from Turkish newspapers during the collection of TurkishBN dataset and it contains 182.3M words.

5. EXPERIMENTS AND RESULTS

5.1. Experimental Setups

In this section we describe the experimental setups in hybrid acoustic models and in end-to-end systems.

5.1.1. ASR Systems with Hybrid Acoustic Models

In this section we describe the language model and acoustic models used in ASR systems with hybrid acoustic models. We built two baseline models one with SLP-1 data and the other one with SLP-1 and SLP-2 data. For adaptation we built source models in two setups, in Setup-1 the source models were trained using only Tedlium dataset, and in Setup-2 they were trained using Tedlium, SLP-1, and SLP-2 datasets. We used the same language model when decoding development and evaluation sets in English Lecture data in all setups. We will start with explaining the language model in the following sub-sections, then, we will provide details of the acoustic models and training procedure for both the baseline acoustic models and the source acoustic models used in the adaptation experiments.

5.1.1.1. Language Model. For the English system, we used the text data explained in the section 4.1.1 for building the language models. The final model used in the experiments is a 4 gram Kneser Ney [54] smoothed language model. We built 3-gram and 4-gram language models with various smoothing approaches. The model yielding the lowest perplexity in the development set was chosen for the experiments.

5.1.1.2. DNN-CE Acoustic Models. The network architecture for the DNN-CE models is shown in Table 5.1. The *Layer* column shows the transformations applied to each layer. The *Input Context* column indicates that the splicing done over time of the

input vectors to that layer. The square bracket implies that all the vectors in-between boundaries are concatenated, and the curly bracket indicates the splicing is done with selection. So an input context of $[-4, 4]$ means that the input vectors are concatenated from $t-4$ to $t+4$. Since the input context is constant through the network in DNN-CE architecture, there is no splicing over time in the hidden layers. The *LDA Transform* layer is an LDA-like transformation matrix constructed with accumulated statistics of the spliced acoustic features and it is kept constant during training. This layer serves as a normalisation layer. The *Affine-PNorm-Norm* layer is composed of an affine transform, p-norm non-linearity function and a normalisation step respectively. The p-norm non-linearity function is applied with $p = 2$ and group size of 5 in baseline models, while in source models the group size is 10. The normalisation step is just an l_2 normalisation applied to hidden activations; $\hat{x} = \frac{x}{\|x\|_2}$. The output dimensions of the hidden layers in the baseline models are shown in Table. 5.1, $(400 - 80)$ indicates the output dimension of the affine transform and p-norm non-linearity function respectively. For the source models, the output dimension of the affine transform is 3000. Since the p-norm is applied with a group size of 10, the output dimensions of the hidden layers in source models become $(3000 - 300)$ with the same notation. While the number of hidden layers in the baseline acoustic models is 4 as shown in Table 5.1, there are 6 hidden layers in the source models. The total number of parameters in the baseline acoustic models is around 560k while in the source models it is around 6.81M. The input features are 13 dimensional MFCC with delta, delta-delta and pitch features which adds up to a 40 dimensional input vector.

The DNN-CE acoustic models is trained for 20 epochs with decaying learning rate starting from 0.02 until it reaches 0.004 in both baseline setups. For the source acoustic models, they are trained for 10 epochs with a mini-batch size of 256 and decaying learning rate starting at 0.001 and it until reaches 0.0001 in both setups.

5.1.1.3. TDNN-CE Acoustic Models. The network architecture for the TDNN-CE models is shown in Table 5.2. The *LDA Transform* layer is obtain in the same way as in DNN-CE acoustic models. The *Affine-ReLU-Norm* layer is composed of an affine

Table 5.1. DNN-CE network architecture for baseline systems.

Layer No	Layer	Input Context	Input Dim	Output Dim
0	LDA Transform	[-4, 4]	360	360
1	Affine-PNorm-Norm	{0}	360	400-80
2	Affine-PNorm-Norm	{0}	80	400-80
3	Affine-PNorm-Norm	{0}	80	400-80
4	Affine-PNorm-Norm	{0}	80	400-80
5	Affine-Softmax	{0}	80	3680

transform matrix followed by a ReLU non linearity and an l_2 normalisation step. The input features to the *LDA Transform* layer is concatenation of 5 consecutive 40 dimensional MFCC feature vectors, $x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2}$, and a 100 dimensional i-vector representation of the acoustic input at time t. The input to a hidden layer H^l with an *Input Context* of {-3, 0, 3} is the concatenation of hidden activation of the layer H^{l-1} at the times $t-3, t, t+3$. The receptive field of the last layer is 21 frames with 13 at the left context and 7 at the right context. The network architectures for baseline acoustic models and source acoustic models are exactly the same in both setups except for the output dimension of the last layer which depends on the number of tied HMM states in the previously trained GMM-HMM acoustic models for the respective models. The total number of parameters for the baseline acoustic model trained with SLP-1 data set is 7.67M and it is 7.86M for the model trained with SLP-1 and SLP-2. For the source model in Setup-1, the total number of parameters is 7.89M and it is 7.85M in Setup-2.

The training sets are augmented with speed and volume perturbation in advance to training. This effectively increases the training examples by 2 fold. All the acoustic models are trained for 2 epochs with decaying learning rate starting at 0.0015 and until it reaches to 0.00015.

5.1.1.4. TDNN-LF-MMI Acoustic Models. The network architecture for the TDNN-LF-MMI acoustic models is shown in Table 5.3. The definition of the layers is the same

Table 5.2. TDNN-CE network architecture.

Layer No	Layer	Input Context	Input Dim	Output Dim
0	LDA Transform	[-2, 2] + [i-vector]	300	300
1	Affine-ReLU-Norm	{0}	300	650
2	Affine-ReLU-Norm	{-1, 0, 1}	1350	650
3	Affine-ReLU-Norm	{-1, 0, 1}	1350	650
4	Affine-ReLU-Norm	{-3, 0, 3}	1350	650
5	Affine-ReLU-Norm	{-6, -3, 0}	1350	650
6	Affine-Softmax	{0}	650	3680

with those in the TDNN-CE model except the hidden layer in the TDNN-LF-MMI model applies batch normalisation instead of l_2 normalisation after ReLU non-linearity step. The batch norm is applied with the following formulation along side the batch dimension,

$$\hat{X} = \frac{X - E[X]}{std(X)}$$

where $E[\cdot]$ is the expectation operation and $std(\cdot)$ is the standard deviation of the input. During inference, the expectation and the standard deviation are replaced with the accumulated statistics of the hidden activations in training. The training for the baseline model with SLP-1 didn't converge despite the efforts of training the model in different sizes. The total parameters for the baseline model trained with SLP-1 and SLP-2 datasets is about 8.37M, for the source models it is about 8.52M in both adaptation setups.

The input to the model is the same with the TDNN-CE model, five 40 dimensional MFCC vectors and i-vector representation of the middle feature vector. The receptive field of the last layer with the given splicing options in Table 5.3 is 27 time frames, 16 on the left and 10 on the right context. The training set is augmented with speed and volume perturbation as in the TDNN-CE training. The baseline model is trained for 12 epochs with a decaying learning rate starting at 0.001 and stopping at 0.0001, and the source model is trained for 4 epochs with the same scheduler.

Table 5.3. TDNN-LF-MMI network architecture.

Layer No	Layer	Input Context	Input Dim	Output Dim
0	LDA Transform	[-2,2]+[i-vector]	300	300
1	Affine-ReLU-BatchNorm	{0}	300	512
2	Affine-ReLU-BatchNorm	{-1, 0, 1}	1536	512
3	Affine-ReLU-BatchNorm	{0}	512	512
4	Affine-ReLU-BatchNorm	{-1, 0, 1}	1536	512
5	Affine-ReLU-BatchNorm	{0}	512	512
6	Affine-ReLU-BatchNorm	{-3, 0, 3}	1536	512
7	Affine-ReLU-BatchNorm	{-3, 0, 3}	1536	512
8	Affine-ReLU-BatchNorm	{-6, -3, 0}	1536	512
9	Affine-ReLU-BatchNorm	{0}	512	512
10	Affine	{0}	512	3152

5.1.2. End-to-End English ASR System

Due to fact that the acoustic model in end-to-end systems require large amounts of training data, we didn't train baseline models in end-to-end framework. We only trained the source models. Even tough it is shown that RNN based acoustic models can achieve good results, they require high computational resources. Considering our limited resources, we only did experiments with transformer based acoustic models. We trained the acoustic model in only Setup-1 settings as explained in Section 3.2. In the following sub-sections we will first describe the details of the speech-transformer acoustic model and the training procedure. Then we will explain the language model used during decoding.

5.1.2.1. Speech-Transformer Acoustic Model. We utilise ESPnet [55] toolkit for testing the speech-transformer based end-to-end system. The encoder part of the acoustic model is composed of a pre-processor module consisting of 2 convolution layers, and a stack of 12 layers of encoder-transformer blocks. The decoder part consists of 6 layers of decoder-transformer blocks. The total number of parameters in the model

is about 330M. The input features are 80 dimensional mel filter banks and the pitch extracted from the acoustic input. Speed perturbation and SpecAugment [56] augmentation methods are applied to the input during training. The model is trained with minimising CTC loss function as explained in Section 2.4.1 with a target of 500 byte pairs. The byte pair model is trained with transcriptions in Tedlium dataset as explained in the section 2.4.2. The model is trained with ADAM optimiser [57] for 50 epochs and the last 10 models are averaged to obtain the final model in both setups.

5.1.2.2. Language Model. The language model that is used during decoding is a LSTM based model trained with byte pair tokens. The Byte Pair Encoder (BPE) is trained with Tedlium training transcriptions targeting 500 tokens. The corpus used for LM training was described in Section 4.1.2. The total number of words in the corpus is about 229.19M and after tokenization with the BPE model, the total number of tokens in the training corpus becomes 478.48M. The transcriptions of the development set in Tedlium dataset is used for validation purpose.

The language model is a 4 layer stacked LSTM model with a unit size of 2048. The network is trained for 2 epochs with SGD optimiser and training takes about 66 hours and 42 minutes. The total number of parameters in the model is about 136.33M.

5.1.2.3. Adapted Language Model. The adapted language model is obtained by fine-tuning the generic LM with the text corpus prepared for English lecture data as explained in Section 4.1.1. The transcriptions of the development set are used for validation purposes during adaptation. The model is fine-tuned until its perplexity on the validation set doesn't decrease for 5 epochs. The model that has the lowest perplexity on the validation set is picked as the final model.

5.1.3. End-to-End Turkish ASR System

We train a speech-transformer model on Turkish Broadcast News dataset to obtain source model in Turkish by utilising EspNet toolkit. The acoustic model has exactly same architecture with the source model in English. A byte pair model with vocabulary size of 500 is trained on the source dataset transcriptions to create targets for the model. As in the source model for English, the training set is augmented with speed and volume perturbation in advance, and SpecAugment augmentation method is applied during training. The network is trained with ADAM optimiser for 50 epochs and the checkpoints in the last 10 epochs are averaged to obtain the final model.

For the LM, we train a 4 layer LSTM model with a unit size of 2048 on Turkish News Corpus [53]. The targets of the model is byte pairs obtained by tokenizing the text corpus with byte pair model trained for the acoustic model. The corpus is a collection of news segments in Turkish newspapers and it contains about 181M words with about 15.13M sentences. After tokenization the total number of tokens becomes 523M. The model is trained for 2 epochs with SGD optimiser and the training takes about 80 hours. The total number of parameters in the model is about 136.33M. We adapt the LM with the reference transcriptions of the SLP-3 adaptation set as we did in the English system.

5.1.4. Adaptation Methods

We adapt source models in both setups by fine-tuning all the parameters with all or some parts of the lecture data. The details of the adaptation procedures are given in the following subsections.

5.1.4.1. Adaptation in Setup-1. We adapt the source model in 3 different settings in Setup-1. In the first setting the SLP-1 lecture dataset is used for adaptation. Because the SLP-1 dataset contains recordings from the same single speaker which the development and evaluation datasets also are comprised of, this adaptation approach is aimed

to test adapting the source model for the speaker.

In the second setting, the SLP-2 dataset is used during adaptation phase. The SLP-2 dataset doesn't contain any recordings from the speaker in the SLP-1 dataset. However, because all the lecture datasets are recorded in the same settings, this adaptation approach aims to test the acoustic condition adaptation scenario.

In the last setting, we use both SLP-1 and SLP-2 for adaptation. In this approach we aim to test the adaptation when both the acoustic and speaker conditioning is applied.

The hybrid-acoustic models in each setting are adapted by fine-tuning all the parameters of the models for 2 epochs with a learning rate 0.0001. For the end-to-end acoustic model, we fine-tune the the whole model with a learning rate of 0.0001 and with an early stopping criteria. The early stopping criteria is that if the accuracy in estimated byte pair tokens of the model does not increase for four consecutive steps the fine-tuning process stops.

5.1.4.2. Adaptation in Setup-2. Since the source model in Setup-2 is already exposed to the adaptation datasets (SLP-1, SLP-2) during training, we do not analyse the acoustic and the speaker adaptation effect separately in this setup. Instead, we finetune the whole model using both the SLP-1 and SLP-2 adaptation sets. We only build source models in this setup with hybrid acoustic models and we finetuned the source models for 2 epochs with a learning rate of 0.00015.

5.2. Results and Discussions

In this section we provide the empirical results for the baseline and adapted ASR systems developed for English and Turkish spoken lecture processing (SLP).

5.2.1. Baseline Results With Hybrid Acoustic Models for English ASR System

Word Error Rates (WERs) of the baseline systems on the evaluation and development sets are given in Table 5.4. From the table we see that the baseline system is benefited from the SLP-2 dataset significantly. Using the SLP-1 and SLP-2 datasets together in acoustic model training improves the results in all models more than %1 in absolute compared to training the models using only the SLP-1 dataset. When we compare the performance of different model architectures, the TDNN-LF-MMI model outperforms the TDNN-CE model and the TDNN-CE model outperforms the DNN-CE model. This is inline with the findings in the literature [6]. All the models perform better on the evaluation set than on the development set. This might be because the corpus used for training the language model contains transcriptions of the Digital Signal Processing and Signals and Systems video lectures collected from MIT open course ware which has a significant overlap in context with the evaluation set. Note that, the evaluation set is composed of video lectures of Signals and System course at MEF University.

Table 5.4. WER with baseline systems.

Training Set	**Test Set**	**DNN-CE**	**TDNN-CE**	**TDNN-LF-MMI**
SLP-1	dev	7.70	7.00	-
	eval	6.18	5.79	-
SLP-1 + SLP-2	dev	6.48	6.24	5.79
	eval	5.21	4.38	4.15

5.2.2. Adaptation Results with Hybrid Acoustic Models for English ASR System

5.2.2.1. Adaptation Results in Setup-1. WER results of the adapted models on the development and evaluation sets are shown in Table 5.5. In the speaker adaptation setting, first row of the table, the best results are obtained by adapting the TDNN-LF-MMI model. This model yields 4.14% WER on the evaluation set and 5.40% WER on

the development set. The TDNN-LF-MMI model outperforms the TDNN-CE acoustic model by about 0.8% in absolute on the evaluation set and 1.5% on the development set. On the other hand, the DNN-CE model yields the highest WERs after adaptation, which are 9.58% on the development and 6.84% on the evaluation set.

Comparing the speaker adaptation performance of the source models with the baseline models, it is apparent that the DNN-CE model does not beat any baseline models. For the TDNN-CE model however, it slightly surpasses the TDNN-CE baseline model trained with SLP-1 and under-performs the TDNN-CE baseline model trained with SLP-1 + SLP-2. For the TDNN-LF-MMI model, the evaluation results are almost on par with the TDNN-LF-MMI baseline model (4.14% vs 4.15%), and the results on development set is better by about 0.4% in absolute (4.40% vs 4.79%). These can be seen by comparing the first row of Table 5.5 with Table 5.4

When the test speaker is not included in the adaptation dataset, the second row in Table 5.5, there is a smaller margin between the performance of the TDNN-LF-MMI and TDNN-CE models compared to their performances in the speaker adaptation setting. The TDNN-LF-MMI model yields 9.76% and 7.80% WERs, and the TDNN-CE model yields 9.90% and 7.37% WERs on the development and the evaluation sets respectively. The performance of the adapted models in this setting does not beat any baseline model, which can be seen by comparing the second row in Table 5.5 with Table 5.4. This suggest that, when the adaptation data is small and the test speaker is excluded, the adaptation approach does not outperform the baseline models.

When both adaptation sets are utilised, third row in Table 5.5, performance of the models follows a similar pattern as in the speaker adaptation setting, such that the TDNN-LF-MMI model achieves the best results with 5.20% and 3.71% WERs on the development and on the evaluation sets. The DNN-CE model, on the other hand, yields the highest WERs among the three models with 9.58% and 6.45% WERs on the development and on the evaluation sets respectively. When comparing the performances of the adapted models with the baseline models, it can be seen that

the adapted DNN-CE model doesn't outperform any baseline model. For the adapted TDNN-CE model, there is a similar situation with the speaker adaptation setting. The adapted TDNN-CE model achieves better results than the baseline TDNN-CE model trained with SLP-1, however it doesn't outperform the baseline model trained with both SLP-1 and SLP-2. It also fails to surpass the TDNN-LF-MMI baseline model. For the adapted TDNN-LF-MMI model, it can be seen that it consistently achieves lower WER than all of the baseline models. These findings suggest that the TDNN-LF-MMI models have greater capability in adaptation than other models which is inline with the findings in the literature [12]. The DNN-CE models fail to surpass baseline models after adaptation in Setup-1.

Table 5.5. WER results with hybrid adapted models.

Training Set	**Adaptation Sets**	**Test Sets**	**DNN CE**	**TDNN CE**	**TDNN LF-MMI**
Tedlium	SLP-1	Dev	9.58	6.91	5.40
		Eval	6.84	4.95	4.14
	SLP-2	Dev	12.59	9.90	9.76
		Eval	8.89	7.37	7.80
	SLP-1 + SLP-2	Dev	9.58	6.44	5.20
		Eval	6.45	4.53	3.71

To investigate the effect of data size in speaker adaptation, we create different subsets from the SLP-1 data set with different sizes and adapted the TDNN-LF-MMI model with each subset. The WER results after adaptation with these subsets are shown in Table 5.6. EE_202 and Math_226 are the courses that SLP-1 dataset is composed of, Subset_1, Subset_2 and Subset_3 are random subsets of SLP-1 with different number of utterances. The number of utterances and the total duration of the utterances are given in the table.

5.2.2.2. Adaptation Results in Setup-2. WER results in Setup-2 is shown in Table 5.7. The performance of the models follows the same pattern as in Setup-1 which is the TDNN-LF-MMI model outperforms other two models and DNN-CE model has

Table 5.6. Speaker adaptation results with different subset of SLP-1.

Adaptation Sets	**number of utterance / total duration**	**Dev**	**Eval**
SLP-1	1073 121m	5.40	4.14
EE_202	503 59m	6.37	4.92
Math_226	570 62m	6.51	4.60
Subset_1	486 54m	6.30	4.61
Subset_2	686 78m	5.69	4.14
Subset_3	886 101m	5.36	4.40

higher WERs than other models. Before adaptation is applied to the source model, the TDNN-LF-MMI model achieves 6.10% and 4.07% WERs on the development and on the evaluation sets respectively and it outperforms the TDNN-CE model with about 0.4% in absolute. After adaptation is applied, the performance of the TDNN-LF-MMI model improves by about 0.7% on the development set by reducing the WER from 6.10% to 5.44% and by about 0.15% on the evaluation set by reducing the WER from 4.07% to 3.92%.

Table 5.7. WER results with hybrid adapted models in Setup-2.

Training Set	**Adaptation Set**	**Test Set**	**DNN CE**	**TDNN CE**	**TDNN LF-MMI**
Tedlium + SLP-1 +SLP-2	-	Dev	7.45	6.60	6.10
		Eval	5.55	4.49	4.07
	SLP1 + SLP-2	Dev	7.28	6.40	5.44
		Eval	4.87	4.11	3.92

The best results obtained with the baseline models and the adapted models in

Setup-1 and Setup-2 are shown in Table 5.8. It is evident that adaptation methods improve the model's performance. It can also be seen that the model in Setup-1 outperforms the model in Setup-2 by about 0.4% in absolute on both the development and the evaluation sets. This maybe because the source model in Setup-2 has already been exposed to adaptation sets during training and consequently there remain less room for improvement.

Table 5.8. Best results in the Baseline, Setup-1 and Setup-2 experiments.

Trainng Set	**Adaptation Set**	**Test Sets**	**TDNN-LF-MMI**
SLP-1 + SLP-2	-	dev	5.79
		eval	4.15
Tedlium	SLP-1 + SLP-2	dev	**5.20**
		eval	**3.71**
Tedlium + SLP-1 + SLP-2	SLP-1 + SLP-2	dev	5.44
		eval	3.92

5.2.3. Adaptation Results in End-to-End English ASR Systems

We only apply Setup-1 adaptation scheme to the end-to-end ASR system. We apply both AM and LM adaptation in end-to-end English ASR systems. For AM adaptation we employ the adaptation in the same settings with the hybrid-models adaptation, namely speaker adaptation, acoustic condition adaptation and both speaker and acoustic condition adaptation. The WER results of the source and the adapted models with no-LM, generic LM and finetuned LM are shown in Table 5.9

The speaker adaptation, second row of Table 5.9, improves the performance by reducing the WER from 30.8% to 5.2% on the development set and from 33.7% to 7.1% on the evaluation set without using an LM during decoding. Before the acoustic model adaption is applied, first row of Table 5.9 the language model adaptation decreases the WER by about 10% in absolute, by reducing the WER on the development set from 23.9% to 14.6% and reducing the WER on the evaluation set from 26.3% to 16.1%. After acoustic model adaptation, the gain from LM adaptation become 0.7% in absolute

for the development set, and by about 2.4% for the evaluation set. The best result is obtained with both acoustic model and language model adaptation, which is 3.9% on the development and 4.1% on the evaluation set. Figure 5.1 shows the accuracy of the acoustic model on the generated byte pair tokens during adaptation. It can be observed that the accuracy on the SLP-1 adaptation set surpasses the accuracy on the development set in 3 epochs and it keeps increasing, while the accuracy on the development set stays in a narrow margin. This suggests that the source model tends to overfit on the adaptation set during adaptation fairly quickly.

Table 5.9. Adaptation results with end-to-end systems.

Acoustic Model	**Test Sets**	**No-LM**	**Generic-LM**	**Finetuned-LM**
Generic AM	dev	30.8	23.9	14.6
	eval	33.7	26.3	16.1
Adapted with SLP-1	dev	5.2	4.6	3.9
	eval	7.1	6.5	4.1
Adapted with SLP-2	dev	17.1	11.7	7.6
	eval	16.7	13.6	6.0
Adapted with SLP-1 + SLP-2	dev	5.6	4.6	3.4
	eval	6.3	5.5	3.3

When only acoustic condition adaptation is applied to the source model, third row in Table 5.9, the reduction in WER is about 13.7% in absolute on the development set and by about 17% in absolute on the evaluation set which is less than speaker adaptation method. However the relative gain with LM adaptation is higher than the relative gain in the speaker adaptation method. This might be because there is more room for improvement with LM adaptation. It can be seen from Figure 5.2, that the accuracy of the acoustic model on the generated byte pair tokens during adaptation follows a different pattern from the the speaker adaptation method. Such that, the accuracy on the adaptation set (SLP-2) does not surpasses the accuracy on the development set and they stay at a similar level. This might be because the SLP-2 dataset is larger in amount by about 105 minutes (4h vs 2h 16m), which prevents the source model to overfit quickly. The adaptation stops because the accuracy on

the development set does not increase for 4 epochs before the model overfits on the adaptation set (SLP-2).

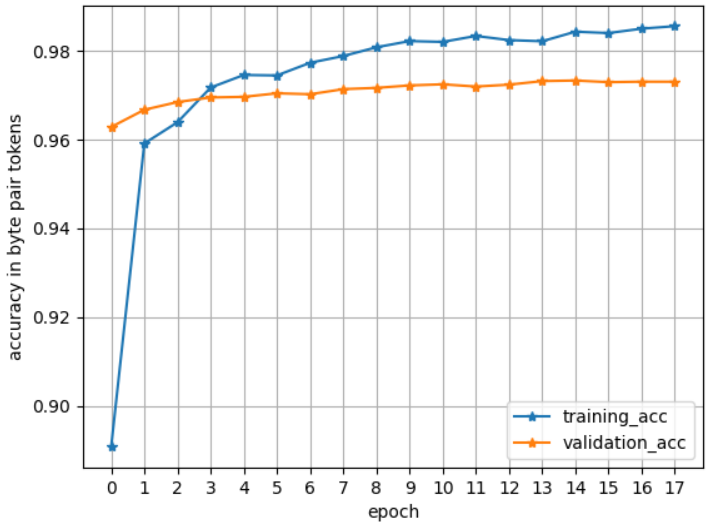

Figure 5.1. Accuracy of the model during adaptation with SLP-1 dataset.

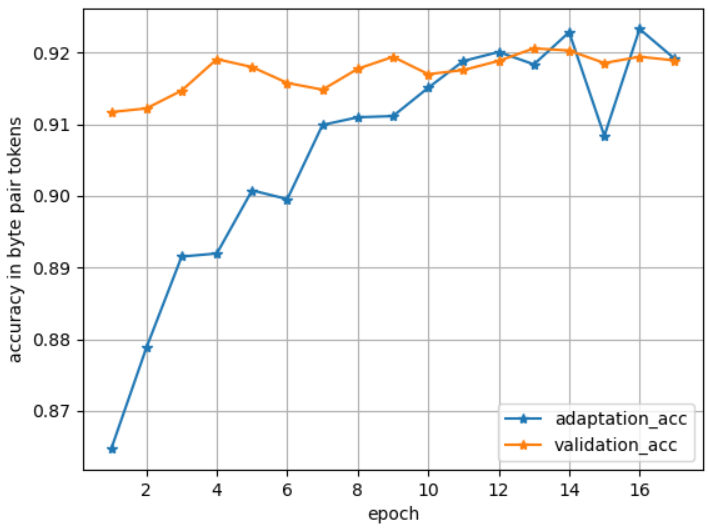

Figure 5.2. Accuracy of the model during adaptation with SLP-2 dataset.

Applying both speaker and acoustic condition adaptation, fourth row in Table 5.9, achieves lower WER than speaker adaptation on evaluation set by about 0.8% and higher WER by about 0.4% on the development set when no LM is used during decoding. This can be interpreted that doing adaptation with both SLP-1 and SLP-2 datasets generalises better than doing adaptation with only SLP-1 dataset based on

the fact that it achieves lower WER on the evaluation set. Adapting the acoustic model with the SLP-1 and SLP-2 dataset using fine-tuned LM in decoding result in the lowest WERs in Table 5.9. The best results are 3.4% and 3.3% WERs respectively on development and evaluation sets. Figure 5.3 show the acoustic model accuracy on the generated byte pairs during acoustic model adaptation. It can be seen that, the model underfits during adaptation and increasing the amount of data can result in even better results.

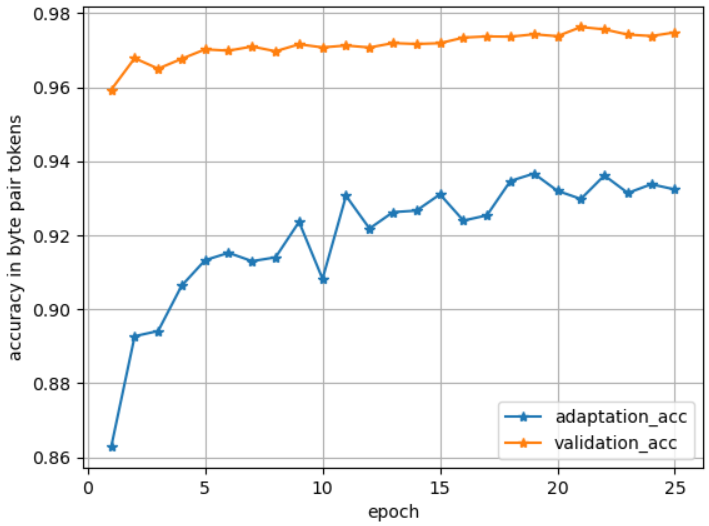

Figure 5.3. Accuracy of the model during adaptation with SLP-1 + SLP-2 datasets.

Comparison of the best results obtained with the ASR systems with hybrid acoustic model and end-to-end ASR system is shown in Table 5.10. It can be seen that while the best results are obtained in end-to-end system, the difference between the performances on the development set is higher than the difference between performances on the evaluation set. This is because the end-to-end model is adapted with the early accuracy criteria on the development set, however the hybrid acoustic models are adapted with fixed number of epochs. The development set is used to chose the weights of the acoustic and the language models during decoding in hybrid acoustic model based ASR systems.

Table 5.10. Hybrid and end-to-end system comparison.

System	Training Set	Adaptation Set	Test sets	Best Results
Hybrid TDNN-LF-MMI	Tedlium	SLP-1 + SLP-2	dev eval	5.2 3.7
Hybrid TDNN-LF-MMI	Tedlium + SLP-1 + SLP-2	SLP-1 + SLP-2	dev eval	5.4 3.9
End-to-End Speech Transformer	Tedlium	SLP-1 + SLP-2	dev eval	**3.4** **3.3**

5.2.4. Adaptation Resulsts in End-to-End Turkish ASR System

We adapt all the parameters of the source model with Turkish spoken lecture data by finetuning the model. Since the amount of Turkish lecture data is much larger than English lecture data we do not stop the adaptation with an early criteria. Instead we let the fine-tuning to run for 50 epochs and average over best 10 models in accuracy. However it can be seen from Figure 5.4 that the validation accuracy start to decrease after 5 epochs of fine-tuning while the training accuracy keeps increasing.

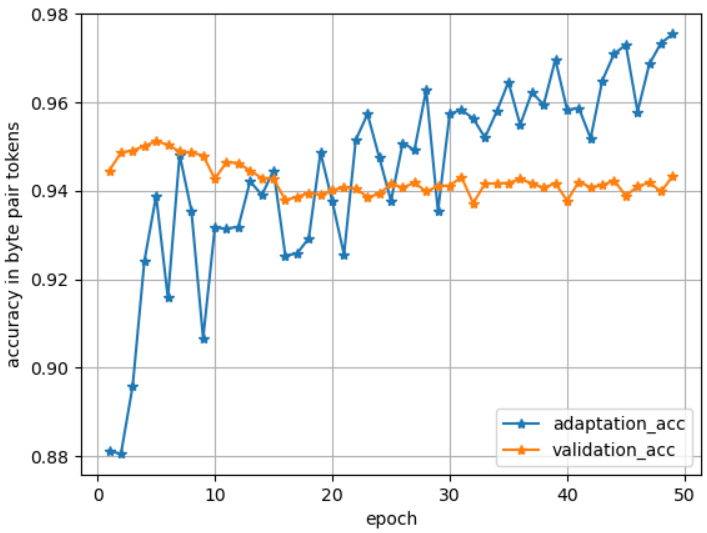

Figure 5.4. Accuracy of the model during adaptation with SLP-3 data set.

Apart from acoustic model adaptation, the language model is also adapted with the transcriptions of the adaptation (SLP-3) dataset. Because the transcription size is

small compared to Turkish News Corpora (183k words vs 183M words) we apply fine tuning with early stopping criteria that if the validation perplexity doesn't increase for 5 epochs the fine-tuning stops. Validation set is generated with transcriptions of the development set.

The WER results of the adapted models are shown in Table 5.11. It can be seen that, the gain in acoustic model adaptation is higher than the gain in language model adaptation. In fact, WER gets slightly higher with adapted language model for the development set. This suggests that the acoustic mismatch between the source dataset and the target dataset is higher than the dissimilarity between the transcription of the SLP-3 dataset and the news corpora.

Table 5.11. Adaptation results for Turkish spoken lecture.

Test Set	**Acoustic Model**	**No-LM**	**Generic-LM**	**Adapted-LM**
Dev	Generic AM	16.8	12.6	12.9
	Adapted AM	13.6	11.8	12.2
Eval	Generic AM	15.3	14.4	13.5
	Adapted AM	11.3	10.4	10.2

The WER results per speaker in the development and adaptation sets are shown in Table 5.12. Speaker-1 has about 5 hours and 20 minutes of speech in the adaptation set. Speaker-2 has about 3 hours and 30 minutes of speech in the adaptation set. The adaptation set does not contain any speech from other speakers.

Table 5.12. WER results in Turkish per speaker.

Test Sets	**Speaker**	**WER**
Dev	Speaker-1	3.8
	Speaker-2	10.7
	Speaker-3	17.9
Eval	Speaker-1	11.0
	Speaker-2	8.57
	Speaker-4	17.7
	Speaker-5	9.8

6. CONCLUSION

Adaptation methods for limited data have been studied extensively in the literature and various approaches have been proposed for different situations. The speech dataset collected from video lectures contains technical terms and it has its own speech style. To apply adaptation methods to spoken lecture data successfully, the dataset used for training the source model needs to be chosen carefully.

In this thesis, we use the Tedlium corpus as the source data set to employ adaptation for English lecture data set. It has similar characteristics with the spoken lecture data set. The speakers in the Tedlium dataset give talks about a topic in a planned manner and in a relatively controlled environment. However it differs from the in-domain English lecture data in terms of the content, and contrary to our in-domain English lecture data, it mostly contains native English speakers. For the Turkish lecture dataset, we utilise the TurkishBN corpus which also has similar characteristics with the Turkish lecture dataset such that it mostly contains planned or read speech. However, unlike the Turkish lecture data, it contains speech from various acoustic environments in addition to the speech in controlled environment generated by the news anchors. It contains only Turkish native speakers which is also the case in the Turkish lecture dataset.

For the English lecture dataset, we apply adaptation in two different setups with ASR systems with hybrid acoustic models, and in only one setup with end-to-end ASR system. In the first setup, Setup-1, we investigate the efficacy of speaker adaptation by adapting the source model with different amount of data coming from the test speaker. We show that with only speaker adaptation 4.1% WER can be achieved on the evaluation set in both ASR systems. By excluding the test speaker and doing adaptation with the remaining adaptation data, which conditions the source model to the changes in the adaptation set like accent, acoustic environment etc., achieves 7.8% WER in the hybrid acoustic models and 6.0% WER in the end-to-end system on the

evaluation set. We obtain the best result when all the available in-domain data for adaptation is used, yielding 3.7% WER with hybrid acoustic model and 3.3% WER with end-to-end system.

In the second setup, Setup-2, we train the source model by using both Tedlium data and English lecture data. The aim of this setup is to implicitly mimic multi-task learning by sharing the model parameters while training the source model on different domain data sets. We do not assign a specific output layer for each domain, instead, by using the same phone set for both domains, we use only one output layer. Even tough the Tedlium dataset is much larger than the in-domain data (118h vs 6h), the source model is able to achieve 4.1% WER on the evaluation set. We find that while adapting the source model trained in Setup-2 improves the performance, it fails to surpass the adapted model obtained in Setup-1.

For Turkish lecture data we do adaptation only with end-to-end system and using Setup-1. The performance of the source model, before adaptation is applied, on the evaluation set is about 16.8% WER when no LM is used during decoding. Here it is important to note that the WER obtained in the same setup with the English end-to-end system is 33.7%. This could be because of the accent mismatch between Tedlium and English lecture data. We observe that adapting the acoustic model improves the results, and this isn't always the case for language model adaptation in end-to-end Turkish ASR system. We also observe that the performance of the model on the development and evaluation sets varies per speaker significantly depending on how much data from each speaker is included in the adaptation set.

REFERENCES

1. Hannun, A., "Sequence Modeling with CTC", *Distill*, 2017, https://distill.pub/2017/ctc.

2. "Creative Commons Attribution CC-BY 4.0", `https://creativecommons.org/licenses/by/4.0/`.

3. Seide, F., G. Li and D. Yu, "Conversational speech transcription using context-dependent deep neural networks", *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

4. Sak, H., A. Senior and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition", *arXiv preprint arXiv:1402.1128*, 2014.

5. Abdel-Hamid, O., A. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu, "Convolutional Neural Networks for Speech Recognition", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22, No. 10, pp. 1533–1545, 2014.

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

7. Saraçlar, M., H. Nock and S. Khudanpur, "Pronunciation modeling by sharing Gaussian densities across phonetic models", *Computer Speech & Language*, Vol. 14, No. 2, pp. 137–160, 2000, `https://www.sciencedirect.com/science/article/pii/S0885230800901402`.

8. Dahl, G. E., D. Yu, L. Deng and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 1, pp. 30–42, 2011.

9. Wang, D. and T. F. Zheng, "Transfer learning for speech and language processing", *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1225–1237, IEEE, 2015.

10. Bengio, Y., F. Bastien, A. Bergeron, N. Boulanger–Lewandowski, T. Breuel, Y. Chherawala, M. Cisse, M. Côté, D. Erhan, J. Eustache, X. Glorot, X. Muller, S. P. Lebeuf, R. Pascanu, S. Rifai, F. Savard and G. Sicard, "Deep Learners Benefit More from Out-of-Distribution Examples", G. Gordon, D. Dunson and M. Dudík (Editors), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Vol. 15 of *Proceedings of Machine Learning Research*, pp. 164–172, JMLR Workshop and Conference Proceedings, Fort Lauderdale, FL, USA, 11–13 Apr 2011, `http://proceedings.mlr.press/v15/bengio11b.html`.

11. Bengio, Y., "Deep Learning of Representations for Unsupervised and Transfer Learning", I. Guyon, G. Dror, V. Lemaire, G. Taylor and D. Silver (Editors), *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, Vol. 27 of *Proceedings of Machine Learning Research*, pp. 17–36, JMLR Workshop and Conference Proceedings, Bellevue, Washington, USA, 02 Jul 2012, `http://proceedings.mlr.press/v27/bengio12a.html`.

12. Ghahremani, P., V. Manohar, H. Hadian, D. Povey and S. Khudanpur, "Investigation of transfer learning for ASR using LF-MMI trained neural networks", *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 279–286, IEEE, 2017.

13. Yosinski, J., J. Clune, Y. Bengio and H. Lipson, "How transferable are features in deep neural networks?", *arXiv preprint arXiv:1411.1792*, 2014.

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

15. Schuppler, B., "Rethinking classification results based on read speech, or: why improvements do not always transfer to other speaking styles", *International Journal of Speech Technology*, Vol. 20, No. 3, pp. 699–713, 2017.

16. Huang, X., A. Acero, H.-W. Hon and R. Reddy, *Spoken language processing: A guide to theory, algorithm, and system development*, Prentice Hall PTR, 2001.

17. Gales, M., "Maximum likelihood linear transformations for HMM-based speech recognition", *Computer Speech & Language*, Vol. 12, No. 2, pp. 75–98, 1998, `https://www.sciencedirect.com/science/article/pii/S0885230898900432`.

18. Matsoukas, S., R. Schwartz, H. Jin and L. Nguyen, "Practical Implementations of Speaker-Adaptive Training", *DARPA Speech Recognition Workshop*, 1997.

19. Bourlard, H. A. and N. Morgan, *Connectionist Speech Recognition: a Hybrid Approach*, Vol. 247, Springer Science & Business Media, 2012.

20. Wang, Y., A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang, C. Fuegen, G. Zweig and M. L. Seltzer, "Transformer-Based Acoustic Modeling for Hybrid Speech Recognition", *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6874–6878, 2020.

21. Le, D., X. Zhang, W. Zheng, C. Fügen, G. Zweig and M. L. Seltzer, "From Senones to Chenones: Tied Context-Dependent Graphemes for Hybrid Speech Recognition", *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 457–464, 2019.

22. Hinton, G., L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups", *IEEE Signal Processing Magazine*, Vol. 29, No. 6, pp. 82–97, 2012.

23. Povey, D., *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. Thesis, University of Cambridge, 2005.

24. Povey, D., V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang and S. Khudanpur, "Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI", *Interspeech 2016*, pp. 2751–2755, 09 2016.

25. Zhang, X., J. Trmal, D. Povey and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks", *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 215–219, IEEE, 2014.

26. Waibel, A., T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang, "Phoneme recognition using time-delay neural networks", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 3, pp. 328–339, 1989.

27. Hadian, H., H. Sameti, D. Povey and S. Khudanpur, "End-to-end Speech Recognition Using Lattice-free MMI", *Proc. Interspeech 2018*, pp. 12–16.

28. Mikolov, T., M. Karafiát, L. Burget, J. Černockỳ and S. Khudanpur, "Recurrent neural network based language model", *Interspeech*, 2010.

29. Chen, S. F. and J. Goodman, "An empirical study of smoothing techniques for language modeling", *Computer Speech & Language*, Vol. 13, No. 4, pp. 359–394, 1999, `https://www.sciencedirect.com/science/article/pii/S0885230899901286`.

30. Kneser, R. and H. Ney, "Improved backing-off for M-gram language modeling", *1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 181–184 vol.1, 1995.

31. Bahdanau, D., J. Chorowski, D. Serdyuk, P. Brakel and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition", *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4945–4949,

2016.

32. Amodei, D., 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", *International Conference on Machine Learning*, pp. 173–182, PMLR, 2016.

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

34. Graves, A., S. Fernández, F. Gomez and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks", *ICML 2006 - Proceedings of the 23rd International Conference on Machine Learning*, Vol. 2006, pp. 369–376, 01 2006.

35. Gage, P., "A new algorithm for data compression", *C Users Journal*, Vol. 12, No. 2, pp. 23–38, 1994.

36. Dong, L., S. Xu and B. Xu, "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition", *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5884–5888, 2018.

37. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need", *arXiv preprint arXiv:1706.03762*, 2017.

38. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proc. NAACL-HLT*, pp. 4171–4186, Minneapolis, MN, USA, 2019.

39. Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettle-

moyer and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach", *Proc. ICLR*, Addis Ababa, Ethiopia, 2020.

40. Ba, J. L., J. R. Kiros and G. E. Hinton, "Layer normalization", *arXiv preprint arXiv:1607.06450*, 2016.

41. Hinton, G. E., S. Osindero and Y.-W. Teh, "A fast learning algorithm for deep belief nets", *Neural Computation*, Vol. 18, No. 7, pp. 1527–1554, 2006.

42. Erhan, D., A. Courville, Y. Bengio and P. Vincent, "Why does unsupervised pretraining help deep learning?", *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 201–208, JMLR Workshop and Conference Proceedings, 2010.

43. Lee, H., P. Pham, Y. Largman and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks", *Advances in Neural Information Processing Systems*, Vol. 22, pp. 1096–1104, 2009.

44. Neto, J., L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals and T. Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system", *International Speech Communication Association*, 1995.

45. Gemello, R., F. Mana, S. Scanzio, P. Laface and R. De Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models", *Speech Communication*, Vol. 49, No. 10-11, pp. 827–835, 2007.

46. Huang, Z., J. Li, S. M. Siniscalchi, I.-F. Chen, J. Wu and C.-H. Lee, "Rapid adaptation for deep neural networks through multi-task learning", *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

47. Huang, Z., S. M. Siniscalchi and C.-H. Lee, "A unified approach to transfer learning of deep neural networks with applications to speaker adaptation in automatic speech recognition", *Neurocomputing*, Vol. 218, pp. 448–459, 2016.

48. Manohar, V., D. Povey and S. Khudanpur, "JHU Kaldi system for Arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning", *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 346–352, 2017.

49. Oppenheim, A., "RES.6-008 Digital Signal Processing. Spring 2011.", , MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

50. Agarwal, A., "6.002 Circuits and Electronics. Spring 2007.", , MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

51. Oppenheim, A., "RES.6-007 Signals and Systems. Spring 2011.", , MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

52. Rousseau, A., P. Deléglise and Y. Esteve, "TED-LIUM: an Automatic Speech Recognition dedicated corpus.", *LREC*, pp. 125–129, 2012.

53. Arisoy, E., D. Can, S. Parlak, H. Sak and M. Saraclar, "Turkish Broadcast News Transcription and Retrieval", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 17, No. 5, pp. 874–883, 2009.

54. Ney, H., U. Essen and R. Kneser, "On structuring probabilistic dependences in stochastic language modelling", *Computer Speech & Language*, Vol. 8, No. 1, pp. 1–38, 1994.

55. Watanabe, S., T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit", *arXiv preprint arXiv:1804.00015*, 2018.

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

57. Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.