ANALYSIS OF SPEAKER ADAPTATION TECHNIQUES IN AUTOMATIC SPEECH RECOGNITION SYSTEMS USING DEEP NEURAL NETWORKS AND GAUSSIAN MIXTURE MODELS

VEERA V RAMA RAO M[1] , KUMAR N[2]

[1]Research Scholar, Dept. of CSE, Vels Institute of Science Technology and Advanced Studies, Chennai, India

[2]Professor, Dept. of CSE, Vels Institute of Science Technology and Advanced Studies, Chennai, India

E-mail: [1]murali.mvv@gmail.com, [2]kumar.se@velsuniv.ac.in

ABSTRACT

The accuracy of automatic speech recognition (ASR) systems can be affected by changes in training and testing environments. In the context of adaptation, narrowing the model-to-dataset discrepancy for a certain speaker or channel is quite effective. Two of the most widely used ASR methods nowadays are deep neural networks and Gaussian mixture models (GMMs). GMM-HMM has been a standard approach in ASR systems for decades. Speaker adaption is especially helpful to AMs in this particular subgroup. Efforts have been made to help this group in several ways. DNN-HMM AMs, on the other hand, have lately beaten GMM-HMM models in ASR tasks. These AMs, on the other hand, frequently have to retrain their accents, which can be difficult for them. As a result, many GMM model modification processes do not apply to DNNs. An explanation of GMM models, as well as ways for increasing speaker adaption, is the primary purpose of this study Domain Adaptation Challenge unsupervised domain adaptation goal data might be collected using DNNs as well (DAC). An out-of-domain system's speaker recognition performance is improved by more than 25 percent by using a DNN trained on data from outside the domain.

Keywords: *Gaussian mixture models, GMM-HMM, Speaker adaptation, AMs, ASR*

1. INTRODUCTION

ASR is the technology that allows human beings to interact with computers by letting them to speak to the computers. ASR advances in the past few years have led to growing adoption of various ASR systems in our daily lives. Most of the ASR systems are now part of our daily lives [1]. Neural network acoustic models have shown significant growth in their ability to adapt. An ASR system with a target speaker and a target channel must use limited amounts of training data from the target acoustic source, while also adapting models (AM) to remove acoustic mismatches when training and then cutting down on the mismatch while testing. Recent improvements in deep learning research have made the developments shown here now achievable.

An ASR system trained with a limited quantity of data adapted to a target speaker or channel seeks to eliminate mismatches between training and testing acoustic settings while also improving the accuracy of the ASR system for a target speaker or channel [2]. The fundamental issue addressed in this study is the inter-speaker variability mismatch, which is sought to be reduced using speaker adaption. However, these ideas can be applied in a far broader range of contexts.

Due to the numerous advantages of GMM-HMM systems, including the fact that GMMs have fewer parameters, training GMM-HMM models can be easily parallelized, and further improvements to the models' performance can be achieved by speaker adaptation training, GMM-HMM systems have been utilized widely because of these characteristics. Nevertheless, GMM-based approaches still have disadvantages, like in the case of GMM-space acoustic feature distributions, which may not always be correct for speech data [3]. Many discriminative models can outperform generative models in classification problems. The DNN-trained systems outperformed GMM-HMM systems on several large vocabulary continuous speech recognition (LVCSR) tasks, and these shortcomings in the GMM-HMM methodology can be addressed by training DNNs using novel deep learning approaches. While these systems still utilise GMMs for training alignment and cluster tree development, DNN-HMM hybrid systems employ them in two other areas as well: initial training alignments and GMM-based cluster tree building [4]. New findings indicate that DNN training can be initiated from the ground up without GMM-HMM system initialization, although there are currently no published works to confirm that cluster trees can be constructed without the use of GMMs.

An expanding area of AI research is the use of deep neural network (DNN) algorithms. Modern ASR systems have mostly replaced traditional Gaussian mixture models (GMMs) with DNN Hidden Markov Models (HMMs). While GMM-HMMs perform well in a variety of ASR tasks, DNN -HMMs outperform them. DNN models, however, are very different from conventional GMM-HMM models [5]. It can be challenging to apply DNN models to machine learning techniques made for GMM-HMM systems, such as MAP, MLLR, and others.

When the models and the facts don't agree for a particular speaker, channel, or other component, adaptation is a good option [6]. The three different types of acoustic models utilised in ASR systems include:

Speaker independent (SI): AM has had extensive training on a training set of acoustic data that has been collected from different speakers, and will generate a pronunciation that is speaker-independent.

Speaker dependent (SD): A narrow target speaker (that is, one person) is being used in AM's training. SD AMs can yield WERs several times lower than SI AMs if provided with sufficient training data.

Speaker adapted (SA) : During the training stage, AM is initially provided with a large dataset of numerous speakers, and then its operation is fine-tuned using a tiny dataset of only one speaker.

1.1 Adaptation types and modes

One way to classify acoustic adaption algorithms is by defining two classifications:

Model-based adaptation: For example, in order to maximise posterior probability or likelihood on the adaption data, it changes the model parameters.

Feature-space adaptation: In most cases, this is acceptable for real-time on-line ASR systems, as it does not require parameter modifications to the acoustic model [7]. Speaker adaptation can be categorized into the following modes depending on the extent to which it utilizes the existing adaptation data.

Batch / off-line / static adaptation: Once all the adaption data have been received, this adaptation is executed. Incremental / on-line / dynamic adaptation: Every time the system makes a new adjustment, more data about the system's evolution is gathered [8]. In order to recognize changes in a speaker, a speaker diarization system is typically used. It relies on the application in selecting the most suitable adaptation mode.

1.2 Normalization

Subsub section has to be in sentense case with no spacing above or blow the srat of it [9]. The goal of normalization is to minimize the impact of a person's unique speech traits. The following are some of the most commonly used methods for normalization.

Cepstral mean and variance normalization (CMVN)
Vocal-Tract-Length Normalization (VTLN)

2. RELATED WORK

A. N.Mishra and his team were responsible for developing an automatic speech recognition system for connected digits that operated without prior knowledge of speakers. To compensate for the characteristics of a clean database, the system incorporated perceptual linear prediction, bark frequency cepstral coefficients, and melfrequency cepstral coefficients. HTK uses HTML. Mishra again helped in developing a technique for isolated digit recognition in Hindi, utilising A.N. Biswas and Astik Biswas' methodology [10]. The HMM classifier is used for feature extraction, and the MFCC method is applied to it. Using both HTK and Mat lab, they carried out several trials, one with clean data and another with noisy data.

To develop a digit recognition system for isolated English digits, [11] developed an HMM and MFCC technique that used a massive library of 50 speakers. HTK is utilised for training and evaluating. Maruti Limkar is developing a system for speech recognition that combines MFCC and DTW (Dynamic time wrapping) algorithm for isolated English digit recognition. In their paper, [12] used HMM and HTK to study American and Chinese spoken English. In [13], they focused on recognising Hindi digits. To collect data in nature, they employed natural noise conditions. In addition, Mohit Dua, along with other researchers, studied digit recognition in Punjabi.

Yang Fan and colleagues found polynomial smoothing effective in their paper (Savitzky-Golay). Use the MACF, as well as the OFTDIM function, and apply it to the calculated correlation distance (W-AMDF). To help differentiate addressed voice waveform, the above methods can be useful. In order to help their customers choose the best pre-signal extraction settings, companies provide experimental findings that show that users can obtain better results using the approach [14]. They also show that employing the tactic increases signal processing efficiency and reduces pitch. There is a definite and unchanging pattern to each person's voice. Changes in temperature and voice pitch are a result of altering the speech signal's pitch.

Md. Akkas Ali et al. unveiled a software that can identify Bangla words automatically. Feature extraction was done using the LPC and Gaussian Mixture Model (GMM).Totally 100 words were recorded in 1,000 trials, resulting in an accuracy rating of 84%.

A system for automatic speech annotation was described by Lucian Georgescu and colleagues that used transcriptions from two complementary ASR systems. In our trials, ASR systems that use DNN and HMM-GMM acoustic models outperformed ASR systems that used HMM-GMM and DNN acoustic models. In this research, the approach developed intends to produce a high-quality annotated speech corpus, all automatically and without supervision. The new corpus was gathered in order to train ASR systems, which in turn would raise the overall acoustic variability of the models and so enhance transcription accuracy.

Humans use language because it is intuitive and practical. Speech recognition (ASR) systems have become widely used and widely available in the modern information age. Due to advancements in ASR system performance and the speed of hardware, technologies, numerous useful applications have emerged, like Alexa, Ok Google, etc. The vast majority of currently available ASR systems are not stable when used in real-world settings. When there is a difference between the training and test conditions, the performance of these systems suffers. Robust ASR systems attempt to effectively deal with this mismatch issue. Both intrinsic (speaker-related) and extrinsic (non-speaker-related) sources of variability contribute to mismatches.

The mismatch owing to non-speaker-related variabilities has received a lot of attention from researchers, but speaker-related variations have received less attention. There is more information than just words in a speech signal, such as the speaker's age, nationality, accent, mood, etc. These data sources are to blame for triggering shifts in linguistic features. In the presence of emotion circumstances, the performances of ASR systems are impacted by the induced source variations. Speaking in an emotional or natural manner is a major difficulty for current ASR systems, which fall under the heading of speaker-related variabilities. As a result, there is a dearth of literature devoted to the development of emotionally resilient ASR systems. There has been no academic standard for ASR in the Telugu language in India, and the difficulty of developing an emotionally robust Telugu ASR only adds to the complexity of the research topic. A standardized Telugu ASR system was therefore attempted.

3. ADAPTATION FOR GMM MODELS

There are numerous GMM-HMM AM algorithms that are highly effective. Here, we go through some of the most frequent methods of adaptation, including MLLR, MAP, and the other strategies, as well as look at some of the most prevalent ways for speaker space.

3.1 Maximum Likelihood Linear Regression(MLLR)

Instead of calculating a sequence of linear transforms, an AM (adaptive model) is developed and used to translate a particular model to an adapted model, the MLLR method (maximum likelihood linear regression) searches for a new model by optimising the probability of adaption data under the transformed Gaussian parameters. This method can be used to analyze variances and/or mean differences. The mean estimate, µb, is determined using the equation:

$$\hat{\mu} = A\mu + b \qquad (1)$$

A is a n ×n transformation matrix (Dimensionality of acoustic feature vectors is known as "n".) and b is a bias vector.

3.2 Maximum a Posteriori (MAP)

3.2.1 Standard MAP approach

One of the most popular methods for acoustic model adaptation is MAP adaptation, often known as Bayesian adaptation.According to certain data O

= (O1, ...,OR), the objective function sought by MAP adaptation is:

$$F_{MAP}(A) = \sum_{r=1}^{R} logp_A(P_A|\emptyset_r)po(A) \quad (2)$$

where φr = φ(Wr),
The MAP adaption strategy employs a prior probability distribution for model parameters. When used to the AM's acoustic model, the term index (m) refers to the index of a Gaussian component in the model. In other words, index (m) stands for the index of a Gaussian component.

$$\widehat{\mu_m} = \frac{\tau\mu_m + \sum_t \gamma_m(t)O_t}{\tau + \sum_t \gamma_m(t)}$$

When working with Gaussian processes, the prior probability of component m, γm(t), and the posterior probability of component m, τ, must be set such that the balance between the maximum likelihood estimate of the mean and its prior value is maintained. Given the relationship between the occupation likelihood of a Gaussian component, given by 3, and the mean-cantered MAP estimate, as seen in Formula 3, the less the occupation likelihood of a Gaussian component, the closer the mean-cantered MAP estimate approaches the SI component mean [15]. With each mean component in the system getting an estimate based on a MAP calculation, the preceding mean, weighting, and adaptation data are used.

Increasing the amount of adaptation data will lead to the system towards speaker-dependent performance in an asymptotic manner. Maximum probability linear regression differs from transformation-based approaches in that, MAP adaptation requires extra data in order to be effective (MLLR).At higher levels of adaption data, MAP surpasses MLLR. It is possible to combine the two adaption approaches to increase performance.

3.2.2 Structural maximum a posteriori (SMAP)

A strategy for boosting MAP forecasts in the absence of enough adaptation data is known as structural maximum a posteriori (SMAP).Using a tree structure, parameters are shared in the SMAP adaption. The first step is to construct a Gaussian distribution tree from Kullback-Leibler divergence as a measurement of how far apart the mixture components are. Each node in the acoustic space has one corresponding Gaussian sound, which is called a leaf. For parent nodes, the parameters are taken as priors, and the method is performed in a cascade manner, from the root node to the leaves.

3.2.3 Vector field smoothing (VFS)

VFS utilises to alleviate one limitation of MAP adaption, while improving its effectiveness even in the face of little adaptation data. In the VFS technique, the speaker-acoustic feature space is assumed to be able to be constantly transferred from one to another. Thus, it's able to tackle the issue of handling unobserved adaptation data. In the VFS method, Gaussian means vectors are taken as input, and the process involves three steps.

1. Estimation of transfer vectors
2. Interpolation of transfer vectors
3. Smoothing of transfer vectors

3.3 Speaker Space Methods

(3)

3.3.1 Cluster adaptive training (CAT)

To find an acoustically comparable cluster of speakers within the training dataset, the concept of speaker clustering is utilised. Since we have an HMM model that matches this cluster, we may use this cluster directly for speech recognition.

Speaker clustering is implemented using the determined speaker distance metric. A number of metrics have attained popularity, among these are the Bhatacharyya distance, probability distance measure, and other. Speaker clustering (SC) can be seen as an extension of cluster adaptive training (CAT). Instead of a statistical model, several model profiles tailored to cluster-specific data are trained on more homogeneous datasets. Gales estimates the speaker model's mean parameters using a linear combination of all the cluster means (2000).All clusters have the same assumed Gaussian component variances and priors. The combination of selected models results in a speaker-specific transform at recognition time.

3.3.2 Eigenvoice-based adaptation

Kuhn et al. provide the same formula, except their basis eigenvoice vectors are multiplied by a tiny weight. In order to find the eigenvoices for a set of supervectors, PCA (principal components analysis) is applied to the system of means of an SD HMM..A subset of the biggest eigenvalues from a set of eigenvoices is picked as a basis set. The components of variance between the reference speakers are represented by the vectors that are orthogonal to each other.

3.4 Speaker Adaptive Training (SAT)

Speaker adaptation (which includes tactics such as Speaker Adaptive Training (SAT)) is intended to better represent a speaker. The approach provided in [16] uses an explicit compensation process to take into account inter-speaker variation in the HMM parameter estimation process. The model is trained first with the use of an initial SI model. The MLLR mean (or fMLLR) transformations are then estimated for each speaker in the training dataset. Finally, using the entire training dataset, the initial models' parameters (mean, variance, and mixture weights) are estimated again with the obtained SD transformations. SAT training reduces the amount of data needed and lowers variance, which provides better likelihoods and reduced variances on the training set. When adaptive testing is used, SAT AM scores often lead to higher results.

4. GMM FRAMEWORK FOR NEURAL NETWORK AMS

The main objective of introducing the GMM framework is to introduce the GMM methodology, which will make it easier to apply GMM adaptation techniques in DNN AMs. Since the 1980s, when they were first used in speech recognition and speaker adaptation, GMM-HMM AMs have been around. A multitude of adaption algorithms designed for GMM AMs have been shown to be rather successful. However, DNNs have made more advancements in voice recognition than GMMs in recent years, and they now outperform GMMs in a number of speech recognition applications. In order to grasp these issues and develop solutions, academics have spent a lot of time on GMM systems. Some adaptation strategies for GMM systems cannot be directly adapted to DNNs. Additionally, although being initially developed for GMM AMs, several feature normalization techniques like VTLN and CMVN have not gained as much traction in recent years. MLLR adaptation is limited to this transformation as there is only one supported feature-space transformation. When the amount of adaptation data is limited, this algorithm performs well; but, when adaptation data rises, it saturates, and is less effective than for example, Bayesian approaches. So, no uniform solution is currently available for efficient transfer of all GMM algorithm implementation strategies to DNN models.Additionally, when GMM and neural network AMs are compared, the models used must be taken into consideration. When it comes to GMM and DNN models, their combination could result in increased ASR efficiency. For this project, we intend to make use of the generalisation ability of GMM adaptive machines, and existing methodologies created for them, in order to apply these approaches to DNN adaptive machines.

4.1 Hybrid DNN-HMM Systems with GMM-Derived Features

Modeling state emission log-likelihoods in a typical GMM-HMM ASR system is done by taking the product of the state emission likelihoods for all linked HMMs in a given coupled state.

$$logP(O_t|s_i) = log\sum_{m=1}^{M} w_{im}N_{im}(O_t|s_i) \quad (4)$$

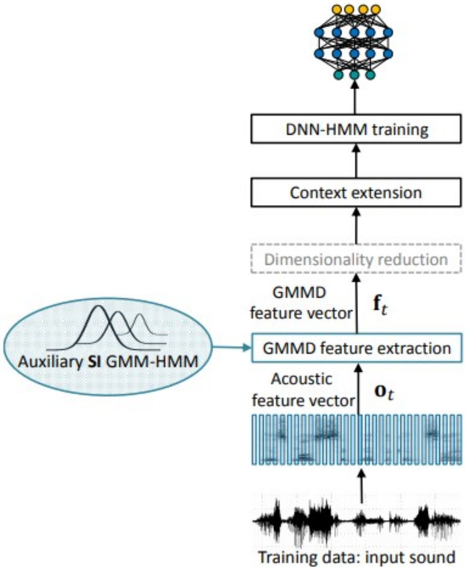

Figure 1: Employing GMMD Features, Training The DNN-HMM Model

If M is the number of Gaussian mixtures for state si and ωim is the mixture weight, then the equation for state si in Gaussian mixture model is M = ωim / ωim + 1.

P(si | ot) are transformed for decoding into pseudo or scaled log-likelihoods, as described by the DNN-HMM system.

$$logP(O_t|s_i) = log\frac{P(s_i|O_t)P(O_t)}{P(s_i)} \,\alpha\, logP(s_i|O_t) - logP(s_i) \quad (5)$$

We'd like to use GMM-enabled DNN models to train. First, we need learn how GMM models are defined so that we may have a better understanding of the concept. The above process describes a feature extraction model (also known as a feature extraction machine model, or just an extraction machine model) that we will refer to as an auxiliary model, and the features that it outputs as derived features.

While developing GMMD for DNNs, there are two primary motivations.

1. NN and GMM models might complement each other and might further increase ASR performance.

2. It is feasible to apply a variety of GMM-HMM algorithm adaptations with this type of features.

To unlock the GMMD features, simply perform the steps listed below. The first step is to extract acoustic feature vectors from the spoken input. Frequency-domain or frequency-based spectral features can be used. In the following step, the collected features can be normalized using Cepstral Mean Normalization (CMN). To do this, we'll utilise a separate HMM-GMM model. With approaches which are used for training this HMM-GMM, it is possible to train it with ML objective function and triphone and monophone states as the basic units.

After deriving the feature vectors for all the states of the auxiliary GMM model, the new GMMD feature vector is calculated using the estimated likelihoods. To get the acoustic feature vector at time t, first do the math. Next, use that result to calculate the new GMMD feature vector, which is computed as follows:

$$f_t = [p_t^1, \dots \dots \dots \dots \dots p_t^n] \qquad (6)$$

When n is equal to the number of variables in the HMM-based model, the additional GMM-HMM model includes states with that value. The auxiliary GMM-HMM AM's number of states determines the size of the likelihood-based feature vector ft. This vector has a length of 1. We can also use PCA, LDA, or HLDA at this stage to reduce the number of dimensions. You do not need to do this step, as demonstrated in Figure 1. In order to acquire a sufficient dimension of the input layer of a DNN, dimensionality reduction must be applied. This GMMD functionality is employed to teach a DNN-HMM AM. Alternatively, GMMD characteristics can be used to train a DNN model, as shown in Figure 1.

4.2 GMMD-related DNN Adaption

In speaker adaption of a SI DNN-HMM model, GMMD features are extracted with an auxiliary SI GMM-HMM model. Any adaptation technique designed for GMM-HMM AMs can be used to implement the adaptation of the SI GMM-HMM model. in this thesis, we will research which speaker adaptation methods are the most popular—MAP and MLLR for DNN-AM speaker adaptation.

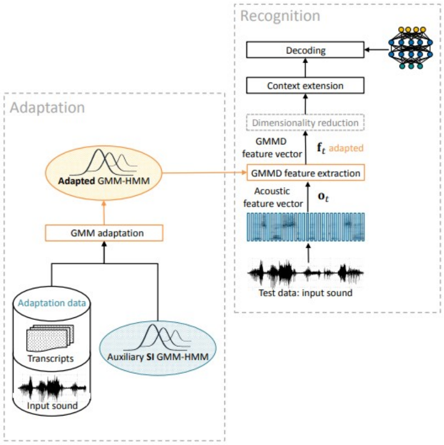

Figure 2: There Is A DNN Using GMMD Features For Machine Learning Adaptation.

In Figure 2 you can see the proposed adaption approach for a DNN model. First, a SI GMM-HMM model adapted to incorporate a speaker is developed, and then a new speaker-adapted (SA) GMM-HMM model is constructed. GMMD features are calculated with this SA GMM-HMM, the HMM for GMMD features. While considered a method rather than a concept, this process can be simply understood as a feature space transformation technique when applied to GMMD-trained HMMs.

4.3 GMMD Feature Analysis

Phoneme posterior based features

A decoder's output lattices are used to obtain a decoder's arc posteriors. This information goes further than the neural network's posterior probability in describing the decoding process. We use these kind of features to determine the acoustic model's adaption performance. Phonemes are a set, and the silent model is. It's necessary to first calculate each time period t and then compute the decoding lattice's confidence score for phoneme phn(1≤n≤N). Using the forward-backward approach, it is possible to calculate this arc posterior probability as follows:

$$P(l|O) = \frac{\sum_{q \in Q_1} P_{qc}(O \backslash q)^{\frac{1}{\delta\lambda}} P_{lm}(w)}{P(O)} \qquad (7)$$

Where λ is the scale factor, qis a path through the lattice corresponding to the word sequence w; Qlis the set of paths passing through arcl;Pac(O|q)

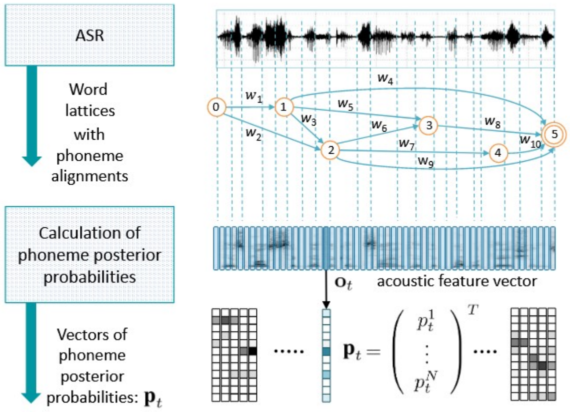
ASR
Word lattices with phoneme alignments
Calculation of phoneme posterior probabilities
Vectors of phoneme posterior probabilities: $\mathbf{p}_t$
w_1
w_2
w_3
w_4
w_5
w_6
w_7
w_8
w_9
w_{10}
0
1
2
3
4
5
o_t acoustic feature vector
$\mathbf{p}_t = \begin{pmatrix} p_t^1 \\ \vdots \\ p_t^N \end{pmatrix}^T$

Figure 3: Sounds following the PPB-based phoneme posterior feature extraction

As a result, each frame of each of the images on the display has a corresponding 3-dimensional vector that represents the chance of the corresponding image to belong to a specific phoneme. When a certain phoneme is not present in the lattice for a certain frame, we set the probability of other phonemes equal to ε. Additionally, we take advantage of the original transcript state index information for the Viterbi alignment.

5. EXPERIMENTS ON AUTOMATIC SPEECH RECOGNITION SYSTEM

5.1 Datasets

The DAC developed by MIT-LL is what we will be using in this experiment. The SRE10 telephone data is used for both enrollment (single cut) and testing (condition 5 extended job). This collection of tests includes 7,169 trials with a specific target and 408,950 trials without a specific target. Domain mismatch during parameter training was investigated using two distinct datasets. There are 36,470 voice clips and 3,790 phone conversations in the pre-SRE10 SRE set, which includes both male and female speakers. The out-of-domain SWB set includes 33,039 voice excerpts from Switchboard-I and II, as well as phone conversations from 3,114 speakers (male and female). However, although both datasets have similar statistics, the SRE set more closely resembles the SRE10 evaluation data. Because of changes in telecommunication systems throughout time, as shown in the analysis in [8], the fundamental cause of the mismatch is apparent. We don't use the in-domain SRE set's labels for our unsupervised adaptation studies; instead, we call it SRE-U.

5.2 DNN Setup

Two p-norm neural networks with power p = 2 were trained with five hidden layers. Except for removal of the fMLLR transform during training and testing, this recipe is quite similar to that given in [8]. DNN-1 is trained on a 100-hour subset of the typical Switchboard data set using P-norms of 4000/400. (LDC97S62). Hidden layers in the DNN2 are trained using the entire 300-hour corpus, which has dimensions 5000/500. Two trigram LMs trained on 3M words of Switchboard training transcripts and 11M words of Fisher English Part 1 transcripts were interpolated to evaluate WERs (LDC2004T19).LDC2002S09, also known as eval2000, has a WER of 16.3% for the larger network and 19.7% for the smaller network in its SWB subset. Table 1 summarizes the network configuration for both networks.

Table 1: Setting up the two DNNs. WER is reported on the Hub5 2000 SWB subset.

Name of System	SWB Train	Hidden layers	WER (%)	Senones	p-norm p/in/out
DNN-1	100h	5	19.7	4295	2/4000/400
DNN-2	300h	5	16.3	9006	2/5000/500

DNN and GMM-based models only differ in how the frame posteriors are computed, and that's it. SSS and ancillary UBM needed for I-vector computation are calculated using DNN-based system's posteriors. The number of senones determines the number of blends of this UBM. As in [12], we make use of mixes with full covariance. 2957 clusters were estimated during the adaption stage (slightly smaller than for the GMM baseline).

5.3 Results

Table 2 lists the evaluation tasks and setups for each system. Quantifying the performance gap caused by the DAC domain mismatch is done using the OOD and IND configurations. There are two unsupervised adaptation studies (UA-LN and UALN-PLDA) that let us determine how much of that gap we can close.Figure 4 shows the DET curves for the three systems in IND and OOD (the baseline and the two DNN setups). In both tasks, DNN-based systems outperformed GMM-based systems. However, the

DNN systems still have a large gap between IND and OOD obligations. The DNN-2 system beats the DNN-1 system. According to this result, an increase in speaker recognition performance can be attributed to more accurate senone classification accuracy (as determined by the WER).As a consequence, it was established that speaker recognition is facilitated by a constant partition of the auditory area depending on phonetic content. Due to the fact that the DNN-based systems outperformed those trained on the rest of the out-of-domain data, it can be concluded that the estimate of the frame posteriors is not sensitive to the DAC's telephony mismatch.

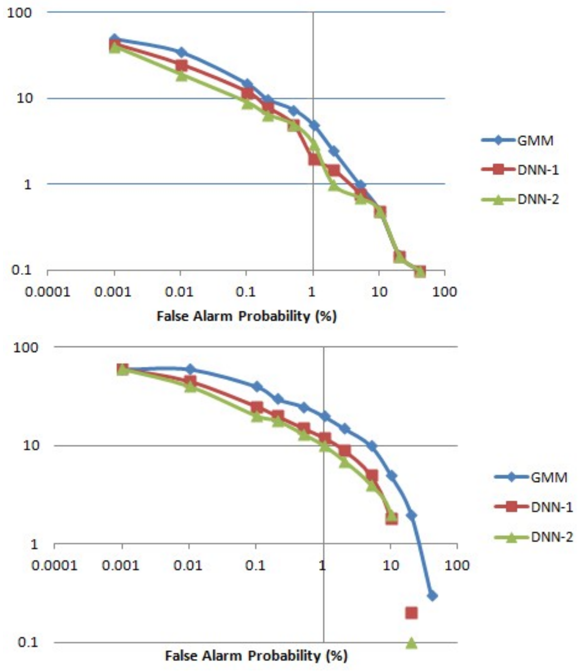

Figure 4: In-Domain And Out-Of-Domain Performance

Table 2: Setup For Each Task

Task	Γ, Λ	UBM, T	m,W
Unsupervised adaptation of length-normalization and PLDA (UA-LN-PLDA)	SRE-U/ SWB	SRE-U	SWB
in-domain (IND)	SRE	SRE-U/ SWB	SRE
out-of-domain (OOD)	SWB	SRE-U	SWB
unsupervised adaptation of length-normalization (UA-LN)	SWB	SWB	SWB

5.4 Limitations

Automatic speech recognition (ASR) systems frequently employ speaker adaption strategies to increase recognition accuracy and robustness for a wide variety of speakers. However, there are bounds to what may be accomplished with speaker adaption approaches. When it comes to automatic speech recognition (ASR) systems, speaker adaptation strategies have a few drawbacks.

Limited data availability:

To successfully adjust the ASR system to an individual speaker, most speaker adaptation algorithms require a large amount of speaker-specific data. For new or rare speakers, or in situations where data privacy considerations limit the availability of speaker-specific data, it may not always be possible to gather a large volume of speaker-specific data in real-world circumstances.

Data variability:

When dealing with speakers who exhibit extremely changeable speech features, such as speakers who significantly alter their voice pitch, speaking rate, or accent during speech, speaker adaption strategies may not be effective. Because of this, it can be difficult to successfully adapt the ASR system to such speakers, as the adaptation might not precisely capture all of the natural variation in their speech.

Speaker disambiguation:

When there are several people talking at once, as in a conversation or a meeting, it can be difficult to keep track of who is saying what. For successful adaptation, it may be necessary to appropriately connect speech fragments with distinct speakers through a process called speaker diarization. However, speaker diarization is not without its flaws, and incorrect adaptation might result from incorrect speaker identification being assigned.

Speaker overfitting:

There is a risk of overfitting when using speaker adaption approaches; this occurs when the ASR system is trained on only one speaker's speech, limiting its ability to generalize to speech from other speakers. Those speakers whose voices are underrepresented in the adaptation data may suffer as a result.

Robustness to environmental factors:

There is no guarantee that incorporating speaker adaption strategies can improve speech recognition accuracy when environmental issues like noise, reverberation, or channel variability are present. In loud or otherwise challenging circumstances, these characteristics may reduce the quality of speaker-specific adaptation data and thus, the efficacy of speaker adaptation strategies.

Computational complexity:
Some methods for adapting a speaker's voice can be rather computationally intensive, necessitating a lot of power and memory to implement. This can hinder the use of ASR systems in real-time or low-latency applications like real-time transcription or voice assistants.

Ethical considerations:
Concerns about privacy and discrimination could arise from the collection and use of speaker data, and any inherent biases in the adaptation process could lead to unfair treatment of certain speakers or groups if they are not adequately addressed.

Finally, while speaker adaptation techniques can help improve ASR system performance, they are not without their drawbacks. These include issues with data availability and variability, speaker disambiguation and overfitting, as well as environmental and computational robustness and ethical concerns. To ensure the successful and ethical usage of ASR systems, it is important to take these restrictions into account during the design and implementation of systems that utilise speaker adaption techniques.

6. CONCLUSION

In GMMD, the capability to alter a DNN-HMM model using DNN-HMM auxiliary GMM-HMM is provided. The updated MAP algorithm for this application will be the subject of our research. The results of the experiments show that the adapted mass balance model is very effective. We looked at the proposed strategy from multiple angles, which used MAP-adapted GMMD features that were derived from the decoding lattices, and discovered that this approach offers the advantages across various layers of the decoding process. The findings of this study reveal a prospective and promising way for future enhancement of the proposed adaption strategy.We examined the effect of transcribed data on the DNN's speaker recognition as a test. Comparing the larger DNN to a smaller one trained on 100 hours of SWB-I data improved speaker detection accuracy (a tiny part of our OOD set). Because of the system's superior speaker recognition, DNNs trained on OOD data can reliably predict the posteriors of senones. OOD and IND statistics differ because telephone networks evolve over time. This implies that DNNs are resistant to inconsistent data. Our results show that the best published results on the DAC's unsupervised domain adaption test come from a system that collects SS using a DNN.

REFERENCES:

[1] Tomashenko, N., Khokhlov, Y., and Esteve, Y. (2016). “On the use of Gaussian mixture model framework to improve speaker adaptation of deep neural network acoustic models”. In *INTERSPEECH*, pp. 3788-3792. [Tomashenko et al.,2016b]

[2] Tomashenko, N., Khokhlov, Y., and Esteve, Y. (2016). “A New Perspective on Combining GMM and DNN Frameworks for Speaker Adaptation”. In International *Conference on Statistical Language and Speech Processing,* SLSP-2016, pp. 120-132. Springer International Publishing. [Tomashenko et al.,2016a]

[3] Tomashenko, N., Khokhlov, Y., Larcher, A., and Estève, Y. (2016). “Exploring GMM-derived features for unsupervised adaptation of deep neural network acoustic models”. In International *Conference on Speech and Computer*, pp. 304-311. *Springer International Publishing*. [Tomashenko et al.,2016d]

[4] Tomashenko N., Vythelingum K., Rousseau A., and Esteve Y. (2016). *LIUM ASRsystems for the 2016 Multi-Genre Broadcast Arabic Challenge* // IEEE Workshop on Spoken Language Technology, SLT-2016, pp. 285–291. [Tomashenko et al.,2016f]

[5] Tomashenko, N. A., and Khokhlov, Y. Y. (2015). “GMM-derived features for effective unsupervised adaptation of deep neural network acoustic models”. In *INTERSPEECH*,pp. 2882-2886. [Tomashenko and Khokhlov,2015]

[6] Tomashenko, N. A., and Khokhlov, Y. Y. (2014).“Speaker adaptation of context dependent deep neural networks based on MAP-adaptation and GMM-derived feature processing”. In *INTERSPEECH*, pp. 2997-3001. [Tomashenko and Khokhlov,2014b]

[7] Khomitsevich O.G., Mendelev V.S., Tomashenko N.A., Rybin S.V., MedennikovI.P.,and Kudubayeva S.A. (2015). “A Bilingual Kazakh-Russian System for Automatic Speech Recognition and Synthesis” // Lecture Notes in Computer Science *(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),*Vol. 9319, pp. 25-33. [Khomitsevich et al.,2015]

[8] Bulgakova E., Sholohov A., Tomashenko N., and Matveev Y. (2015). “Speaker Verification Using Spectral and Durational Segmental Characteristics”// *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 9319, pp. 397-404. [Bulgakova et al.,2015a]

[9] Tomashenko N., and Khokhlov Y. (2014). “Speaking Rate Estimation Based on Deep Neural Networks”. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* Vol.[Tomashenko and Khokhlov,2014c]

[10] Chernykh G., Korenevsky M., Levin K., Ponomareva I., and Tomashenko N. (2014). “State Level Control for Acoustic Model Training” // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*), Vol. 8773, No. LNAI, pp. 435–442. [Chernykh et al.,2014b]

[11] Levin, K., Ponomareva, I., Bulusheva, A., Chernykh, G., Medennikov, I., Merkin, N.,Prudnikov, A., and Tomashenko, N. (2014). “Automated closed captioning for Russian live broadcasting”. In *INTERSPEECH* (pp. 1438-1442). [Levin et al.,2014]

[12] Tomashenko, N. A., and Khokhlov, Y. Y. (2013). “Fast Algorithm for Automatic Alignment of Speech and Imperfect Text Data”. In Speech and Computer: 15th *International Conference, SPECOM 2013*, September 1-5, 2013, Pilsen, Czech Republic, *Proceedings*. Vol. 8113, pp. 146–153. *Springer*. [Tomashenko and Khokhlov,2013]

[13] Khokhlov, Y., and Tomashenko, N. (2011). “Speech recognition performance evaluation for LVCSR system”. In Speech and Computer: 14th *International Conference,* SPECOM2011. *Kazan* pp. 129-135. [Khokhlov and Tomashenko,2011]

[14] Khokhlov Y., Tomashenko N., Medennikov I., and Romanenko A. (2017). “Fast and Accurate OOV Decoder on High-Level Features”. In *INTERSPEECH.* pp 2884-2888.[Khokhlov et al.,2017b]

[15] Khokhlov Y., Medennikov I., Romanenko A, Mendelev V., Korenevsky M., PrudnikovA., Tomashenko N., and Zatvornitsky A. (2017). “The STC Keyword Search System for OpenKWS” 2016 Evaluation. In *INTERSPEECH*. pp 3602-3606. [Khokhlov et al.,2017a]

[16] Medennikov, I., Romanenko, A., Prudnikov, A., Mendelev, V., Khokhlov, Y., Korenevsky, M., Tomashenko N., and Zatvornitskiy, A. (2017). “Acoustic Modeling in the STC Keyword Search System for OpenKWS” 2016 Evaluation. In *International Conference on Speech and Computer*. pp. 76-86. *Springer International Publishing* [medennikov etal 2017]