BIAS MITIGATION AGAINST NON-NATIVE SPEAKERS IN DUTCH ASR

Master Thesis

ter verkrijging van de graad van master
aan de Technische Universiteit Delft

door

Yixuan ZHANG

student aan Faculteit Elektrotechniek, Wiskunde & Informatica,
Technische Universiteit Delft, Delft, Nederland

Dit proefschrift is goedgekeurd door de

promotor: dr. O.E. Scharenborg

Samenstelling promotiecommissie:

Dr. ir. J. Dauwels,	Technische Universiteit Delft
Dr. C. Varon,	Technische Universiteit Delft
Dr. T. Patel	Technische Universiteit Delft

Keywords: Automatic speech recognition, bias, transfer learning, data augmentation

"Is that everything? It seemed like he said quite a bit more than that."

Lost in Translation, 2003

CONTENTS

ABSTRACT

One of the most important problems that needs tackling for wide deployment of Automatic Speech Recognition (ASR) is the bias in ASR, i.e., ASRs tend to generate more accurate predictions for certain speaker groups while making more errors on speech from others. In this thesis, we aim to reduce bias against non-native speakers of Dutch compared to native Dutch speakers. Typically, an important source of bias is insufficient training data. We therefore investigate employing three different data augmentation techniques to increase the amount of non-native accented Dutch training data, i.e., speed and volume perturbation and pitch shift, and using these for two transfer learning techniques: model fine-tuning and multi-task learning, to reduce bias in a state-of-the-art hybrid HMM-DNN Kaldi-based ASR system. Experimental results on read speech and human-computer interaction (HMI) speech showed that although individual data augmentation techniques did not always yield an improved recognition performance, the combination of all three data augmentation techniques did. Importantly, bias was reduced by more than 18% absolute compared to the baseline system for read speech when applying pitch shift data augmentation and multi-task training, and by more than 7% for HMI speech when applying all three data augmentation techniques during fine-tuning, while improving recognition accuracy of both the native and non-native Dutch speech.

1

INTRODUCTION

1.1. MOTIVATION

Automatic Speech Recognition (ASR) systems have evolved from discriminating among isolated digits to recognizing telephone-quality, spontaneous speech, allowing for a growing number of practical applications in various sectors [1]. However, serious challenges that ASR systems are facing in almost every stage of the speech recognition process never fade away. As deep learning becomes more and more widely used in modern industry to solve statistical problems, researchers started to pay attention to the unfairness the deep learning algorithms are subject to, although deep neural networks (DNNs) are often considered a harbour of objectivity [6]. State-of-the-art (SOTA) ASR systems are built on the basis of DNNs, therefore, the fairness concerns are also mounting in recent ASR-related study. Various works have shown that speech variability due to gender [3, 9, 18], age [2], speech impairment [14], regional accents [9], racial disparities [19, 11], and non-native accents [22, 20] lead to recognition accuracy gaps among speaker groups. There are many reasons for these biases to occur, such as imbalanced training data sets, vocal characteristics of certain speaker groups, mismatch between the test data and the training data, and specific architectures and algorithms used during ASR system development [6].

As globalisation emerges, more and more people do not only speak their mother tongues. People learn new languages with different motivations - some are tourists, some are immigrants, some are interested in foreign cultures, some want to acquire certain knowledge or want to develop extra skills. At the same time, the popularity of using ASR systems as a natural interface is increasing with the maturity of speech recognition technology, whereas most non-native speakers who try to use ASR applications will probably be disappointed by their performance [20]. As seen in [6], the average word error rate (WER) difference of the SOTA Dutch ASR system is the biggest between native speakers and non-native speakers (absolute 36.2% and 47.5% in read and human machine interaction (HMI) speech). Therefore, this thesis primarily focuses on mitigating the bias, i.e. the WER difference, between native and non-native Dutch speech.

1

Previous research suggests that the most major cause of the bias against accented speech is the lack of accented speech data [**Vu2014Impr ovingAP**, 12, 21], and various attempts have been made to reduce the absolute recognition WER of accented speech. If we would like to offset the negative influence on the recognition accuracy of accented speech exerted by this lack, the methods widely adopted can be categorized into:

1. Increasing the amount of non-native speech;

2. Improving the learning efficiency of the model when learning from the limited non-native speech resources.

Data augmentation, which refers to methods for constructing iterative optimization or sampling algorithms using unobserved data or latent variables [5], is proven effective in handling data sparsity [4, 10] and enhancing the performance of deep-neural-network-based acoustic models on accented speech recognition tasks [17]. However, the application of data augmentation targeting reducing the bias between native and non-native speech has been very limited. The previous research mentioned above mainly focused on augmenting the original data, wishing for improvement in recognition accuracy, not in reducing bias. Therefore, this research investigates the effect of applying data augmentation techniques on non-native speech data as well, for the bias reduction purpose.

Traditional machine learning techniques try to learn each task from scratch, while transfer learning techniques try to transfer the knowledge from some previous tasks to a target task when the latter has fewer high-quality training data [15]. This characteristic of transfer learning allows it to be a strong candidate for solving the data sparsity problem in ASR of accented speech, and its effect in reducing the WER of target tasks are proven remarkable by a number of previous research [8, 17, 7, 13]. Therefore, it comes naturally that transfer learning could be employed to see if it reduces the bias against non-native speech. Considering whether the data from source domain and target domain is labeled or not, transfer learning techniques can be further divided into self-taught learning, fine-tuning, multi-task learning, domain adaptation, and self-taught clustering [15]. Within the scope of this research, since the corpora to be used in this research consist of labeled data only, fine-tuning and multi-task learning which are set for labeled data are employed. How these techniques differ and how the categorization of them is done will be introduced in detail later in 2.

1.2. RESEARCH QUESTIONS

In this research, I aim to reduce the bias against non-native speakers in Dutch ASR systems using data augmentation techniques and transfer learning strategies. The research questions are thus as follows:

- Could data augmentation help reduce the bias against non-native accented speech in ASR systems without causing harm to the recognition accuracy of native speech?

- Could fine-tuning and multi-task learning be effective in reducing bias against non-native accented speech without causing harm to the recognition accuracy of native speech, when compared with standard training methods?

1.3. OVERVIEW

To answer the research questions above, experiments have been conducted accordingly. Kaldi [16], an open-source speech recognition toolkit written in C++, is used to build a SOTA Dutch ASR model which serves as the baseline. Then the application of data augmentation and transfer learning strategies are performed to investigate their effect in reaching the goal of answering the questions. The structure of the thesis is: In chapter 2, we introduce the working principle of an ASR system, and the mechanism of Kaldi toolkit; in chapter 3, we take a closer look at the actual approaches when conducting the experiments, and the pipeline of building an ASR system with Kaldi; in chapter 4, our experiments and the corresponding results are presented along with some discussion; in the final chapter, chapter 5, a brief conclusion about this research project is given, along with a possible future work plan.

BIBLIOGRAPHY

[1] Mohamed Abdel-rahman. "Deep Neural Network Acoustic Models for ASR". PhD thesis. University of Toronto, 2014.

[2] Mohammad Abushariah and Majdi Sawalha. "The effects of speakers' gender, age, and region on overall performance of Arabic automatic speech recognition systems using the phonetically rich and balanced Modern Standard Arabic speech corpus". In: Jan. 2013.

[3] Martine Adda-Decker and Lori Lamel. "Do speech recognizers prefer female speakers?" In: Sept. 2005, pp. 2205–2208. DOI: 10.21437/Interspeech.2005-699.

[4] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. "Data Augmentation for Deep Neural Network Acoustic Modeling". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.9 (2015), pp. 1469–1477. DOI: 10.1109/TASLP.2015.2438544.

[5] David A van Dyk and Xiao-Li Meng. "The Art of Data Augmentation". In: *Journal of Computational and Graphical Statistics* 10.1 (2001), pp. 1–50. DOI: 10.1198/10618600152418584. eprint: https://doi.org/10.1198/10618600152418584. URL: https://doi.org/10.1198/10618600152418584.

[6] Siyuan Feng et al. *Quantifying Bias in Automatic Speech Recognition*. 2021. DOI: 10.48550/ARXIV.2103.15122. URL: https://arxiv.org/abs/2103.15122.

[7] Pegah Ghahremani et al. "Investigation of transfer learning for ASR using LF-MMI trained neural networks". In: *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. 2017, pp. 279–286. DOI: 10.1109/ASRU.2017.8268947.

[8] Abhinav Jain, Minali Upreti, and Preethi Jyothi. "Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning". In: *INTERSPEECH*. 2018.

[9] Liu Wai Kat and P. Fung. "Fast accent identification and accented speech recognition". In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*. Vol. 1. 1999, 221–224 vol.1. DOI: 10.1109/ICASSP.1999.758102.

[10] Tom Ko et al. "Audio augmentation for speech recognition". In: *Proc. INTERSPEECH*. 2015, pp. 3586–3589. DOI: 10.21437/Interspeech.2015-711.

[11] Allison Koenecke et al. "Racial disparities in automated speech recognition". In: *Proceedings of the National Academy of Sciences* 117.14 (2020), pp. 7684–7689. DOI: 10.1073/pnas.1915768117. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1915768117. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1915768117.

[12] Allison Koenecke et al. “Racial disparities in automated speech recognition”. In: *Proceedings of the National Academy of Sciences* 117.14 (2020), pp. 7684–7689. ISSN: 0027-8424. DOI: 10.1073/pnas.1915768117. eprint: https://www.pnas.org/content/117/14/7684.full.pdf. URL: https://www.pnas.org/content/117/14/7684.

[13] Julius Kunze et al. *Transfer Learning for Speech Recognition on a Budget*. 2017. DOI: 10.48550/ARXIV.1706.00290. URL: https://arxiv.org/abs/1706.00290.

[14] Laureano Moro-Velázquez et al. “Study of the Performance of Automatic Speech Recognition Systems in Speakers with Parkinson’s Disease”. In: Sept. 2019, pp. 3875–3879. DOI: 10.21437/Interspeech.2019-2993.

[15] Sinno Jialin Pan and Qiang Yang. “A Survey on Transfer Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.

[16] Daniel Povey et al. “The kaldi speech recognition toolkit”. In: *In IEEE 2011 workshop*. 2011.

[17] Xian Shi et al. *The Accented English Speech Recognition Challenge 2020: Open Datasets, Tracks, Baselines, Results and Methods*. 2021. arXiv: 2102.10233 [cs.SD].

[18] Rachael Tatman. “Gender and Dialect Bias in YouTube’s Automatic Captions”. In: *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 53–59. DOI: 10.18653/v1/W17-1606. URL: https://aclanthology.org/W17-1606.

[19] Rachael Tatman and Conner Kasten. “Effects of Talker Dialect, Gender Race on Accuracy of Bing Speech and YouTube Automatic Captions”. In: Aug. 2017, pp. 934–938. DOI: 10.21437/Interspeech.2017-1746.

[20] Tan Tien Ping. “Automatic Speech Recognition for Non-Native Speakers”. Theses. Universit ’e Joseph-Fourier - Grenoble I, July 2008. URL: https://tel.archives-ouvertes.fr/tel-00294973.

[21] Thibault Viglino, Petr Motlícek, and Milos Cernak. “End-to-End Accented Speech Recognition”. In: *INTERSPEECH*. 2019.

[22] Yunhan Wu et al. “See what I’m saying? Comparing Intelligent Personal Assistant use for Native and Non-Native Language Speakers”. English. In: *Mobile HCI 2020*. United States: Association for Computing Machinery (ACM), Oct. 2020. DOI: 10.1145/3379503.3403563.

2

BACKGROUND

This chapter provides basic background information about ASR, which starts from an overview of the ASR system, and then describes the building process of acoustic model and language model. Furthermore, how deep learning plays a role in modern ASR systems, how the training strategies - fine-tuning and multi-task learning - exert influence on the tasks, and how the Kaldi toolkit does the modeling in a slightly different fashion compared with the theory, are introduced. Last but not least, the evaluation metrics used in this work is presented.

2.1. AUTOMATIC SPEECH RECOGNITION

ASR lies within the field of pattern recognition. As its name suggests, an ASR aims to recognise a given input speech signal and output the most likely word sequence corresponding to it. If a sequence of acoustic feature vectors $\boldsymbol{X} = (x_1, x_2, x_3, ...)$ is extracted from input speech signal, and $\boldsymbol{W}$ denotes a word sequence, the most likely word sequence $\boldsymbol{W}^*$ can be computed by:

$$\boldsymbol{W}^* = argmax_W P(W|\boldsymbol{X}) \tag{2.1}$$

applying Bayes' Theorem to simplify the calculation:

$$\boldsymbol{W}^* = argmax_W P(W|\boldsymbol{X}) = argmax_W P(\boldsymbol{X}|W)P(W) \tag{2.2}$$

where $P(\boldsymbol{X}|W)$ is the likelihood of the feature vector sequence extracted from the given word sequence $\boldsymbol{W}$. Since the prior probability of the feature vectors $P(X)$ is the same for all possible word sequences, it is ignored during the maximisation of $argmax_W P(W|\boldsymbol{X})$. $P(\boldsymbol{X}|W)$ predicts what speech units each speech segment corresponds to, so it is called the acoustic model. $P(W)$ is the likelihood of the word sequence, which is computed from a probability distribution over sequences of words, i.e. the language model.

The architecture of ASR is shown in Figure 2.1. Firstly, the input signal is pre-processed to extract features, which derive the characteristics of speech. In this process, the speech

signal is divided into small segments, and the features are extracted from each segment. In this research, Mel-Frequency Cepstral Coefficients (MFCCs) are employed as the features, which can be extracted from the original sound clip through windowing, applying discrete Fourier transform (DFT), taking the log of the magnitude, and then warping the frequencies on the Mel scale, lastly performing the inverse discrete consine transform (IDCT) of the log filterbank energies. The next step is to search for the word sequences with the highest probability using the acoustic model, the language model, and a lexicon that maps the words to phones. In hybrid ASR models, the Viterbi algorithm is commonly used to carry out this decoding process.

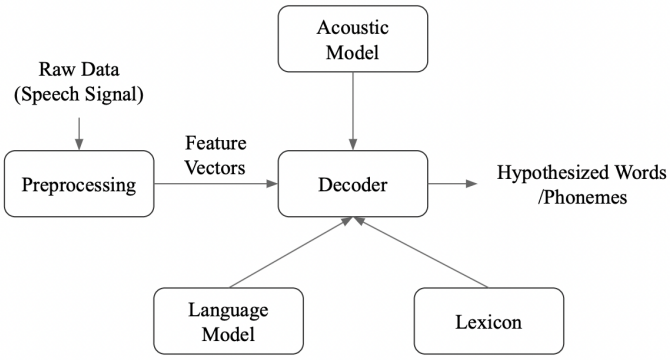

Figure 2.1: Process of generating an ASR system

2.1.1. ACOUSTIC MODEL

The acoustic model aims to model a sequence of features given a sequence of words (phones). The transition between phones and its corresponding features, say, MFCCs, can be modeled with the Hidden Markov Model (HMM); the distribution of features can be modeled with a Gaussian Mixture Model (GMM), which is used to estimate the likelihoods that act as the HMM state observation (features in our case) probabilities. SOTA acoustic model for Dutch is built with deep neural network (DNN) [20]. DNN replaces the position of GMM, which leads to the development of models surpassing the accuracy of GMM-based models.

HIDDEN MARKOV MODEL

HMMs provide a simple and effective framework for modelling the units of speech (e.g. phones, words) as sequences of states. As shown in Figure 2.2, the top row of nodes are internal states, which generally represent phones in a lexicon, while the lower row of nodes represent observable measurements, such as MFCC features. The probability of observing an observable feature x_t given an internal state i is called the emission probability. The probability of transitioning from one internal state i to another state j is called the transition probability, and can be denoted as a_{ij}. On entering a state x_t, a feature vector is generated under the emission probability $b_i(x_t)$ associated with the state being entered [8].

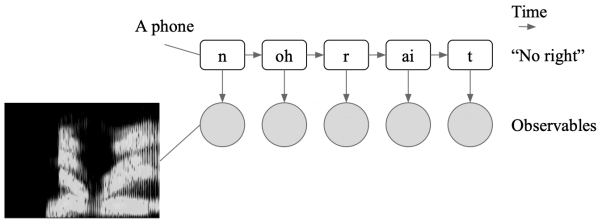

Figure 2.2: Hidden Markov Model

GAUSSIAN MIXTURE MODEL

GMMs model the distribution of the emission probability $b_i(x_t)$ mentioned above. By assuming that $b_i(x_t)$ follows a Gaussian distribution, the aforementioned likelihood $P(X|W)$ can be modelled as

$$P(X|W) = a_{01} \prod_{t=1}^{T} b_t(x_t) a_{tt+1} \tag{2.3}$$

where T is the total number of states. How the GMM is combined with HMM is illustrated in Figure 2.3.

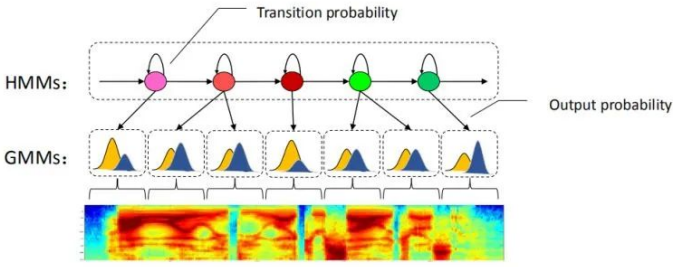

Figure 2.3: GMM-HMM acoustic model

DEEP NEURAL NETWORK

Researchers found that in phoneme recognition, applying neural nets (NN) gives better recognition performance than traditional GMM-HMM models at the beginning of 90s [19]. This suggests NN has the potential to construct better acoustic models than GMM. The increase in computational power has also enabled more powerful deep neural networks (DNNs) - in particular time-delay neural networks (TDNNs) [16], convolutional neural networks (CNNs) [1], long short-term memory (LSTM) recurrent neural networks (RNNs) [10], and bidirectional LSTMs [9] - to be employed for the role of GMM[3]. How DNNs are combined with HMMs is illustrated in Figure 2.4. In this section, starting from a basic introduction of deep learning, the application of DNN in acoustic modelling is introduced.

Machine learning refers artificial intelligence (AI) algorithms which identify patterns from mass data, and make predictions. Deep learning is a subcategory of machine learn-

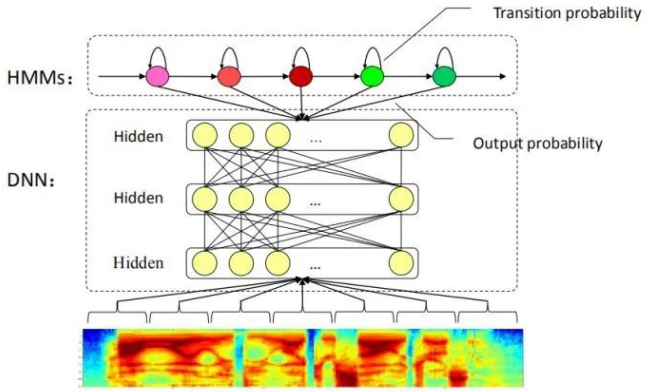

Figure 2.4: DNN-HMM acoustic model

ing, where by 'deep' it means that deep learning craves more data than machine learning does. Deep learning algorithms are built on the basis of neural network (NN) layers, each of which comprises small units named neurons that perform non-linear transformations. Feedforward neural networks (FNNs) are the essential deep learning models, where 'feedforward' indicates that the data flow from the input goes straight to the output without data flowing backward. The goal of a feedforward network is to approximate some function f^*. For example, for a classifier, $y = f^*(x)$ maps an input x to a category y. A feedforward network defines a linear mapping $y = f(x;\theta)$ and learns the value of the parameters θ that result in the best function approximation. Within the scope of acoustic modelling, the DNN maps the emission probabilities of certain phones to the corresponding features. 2.5 illustrates the structure of a basic FNN.

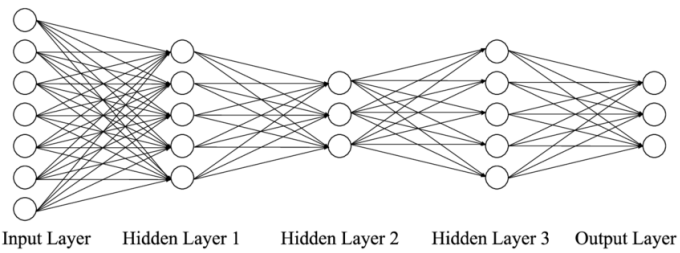

Figure 2.5: Feedforward neural network architecture[11]

To make the mapping function $f^*(x)$ close to the actual function $f(x)$ as much as possible, activation functions are used to introduce non-linearity to linear FNNs, and loss functions are used to quantify how far the predicted outputs of the network are from the actual scenarios. Common loss functions include Mean Squared Error (MSE) for regression problems and cross-entropy for classification problems. In general, cross-entropy and its modified versions are commonly used in ASR tasks. Because of the non-linearity caused by activation functions, many loss functions applied in neural networks

become non-convex. Therefore, gradient-based optimizers such as stochastic gradient descent are used in training deep neural networks, which minimizes the loss functions iteratively. The gradient of the loss function with respect to the parameters θ is computed in each step, then θ is updated in the opposite direction of the gradient, hence minimising the value of the loss function. Some widely adopted gradient-descent algorithms are, to name a few, stochastic gradient descent, momentum, root mean squared propagation (RMSprop), and adaptive moment estimation (Adam).

Convolutional neural networks (CNNs) [14] are originally proposed for image recognition problems with their unique advantage of capturing spatial structures of images, assuming nearby pixels are correlated with each other. This characteristic allows it to be applied in speech recognition tasks, since the features of speech can also be visualised as images. In [14], a CNN-HMM hybrid acoustic model was applied in the ASR system, which had better performance in speech recognition tasks than GMM-HMM based system.

Time Delay Neural Networks (TDNNs) [16] are one-dimensional CNNs, also perform relatively well and efficiently as acoustic models in the field of speech recognition [7]. There are many popular variations of TDNN in its family, including time delay neural network factorisation (TDNNF) [18], time delay neural neural network with long short-term memory (TDNN-LSTM) [17], and time delay neural network with bidirectional long short-term memory (TDNN-BLSTM) [5]. TDNNF has been picked as the baseline acoustic model due to its relative low bias against non-native speech of Dutch compared with TDNN-LSTM and TDNN-BLSTM [7].

The TDNNF, which will be applied in the experiments later, is inspired by the SVD (Singular Value Decomposition) method of reducing network parameters, which has become popular in recent years. SVD factorises the network weight matrix into two smaller matrices and discarding the smaller singular values [18].

Povey [18] applied SVD to TDNN and added a series of strategies such as L2 regulation, "floating" semi-orthogonal constraint, 3-stage convolution per-layer, dropout, skipping connections, etc. The result is the TDNNF, which is structurally the same as a TDNN whose layers have been compressed via SVD, but is trained from a random start with one of the two factors of each matrix constrained to be semi-orthogonal. This boosts the computational efficiency of TDNNs and performs equally well as TDNN-LSTM hybrid systems. The overall architecture of a TDNNF is illustrated in Figure 3.1:

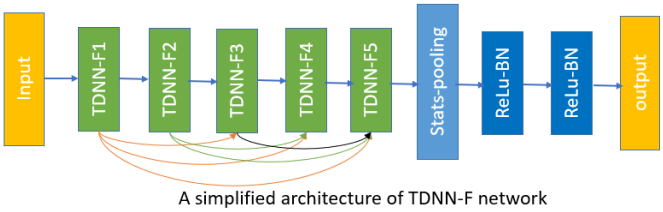

Figure 2.6: TDNNF

where the difference between an original TDNN block and TDNNF block can be

viewed intuitively in Figure 2.7.

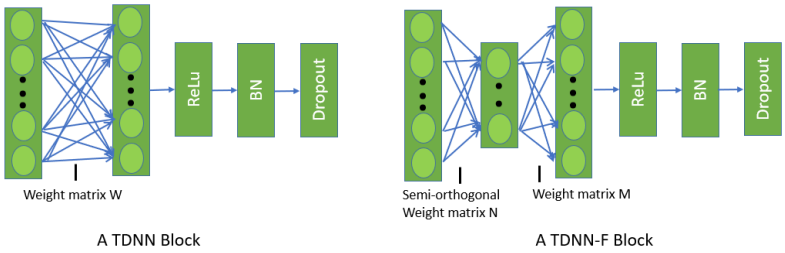

Figure 2.7: TDNNF in comparison with TDNN

As shown above, the major difference between TDNNF and TDNN is that the original weight matrix W is decomposed into two matrices M and N, and N is constrained to be semi-orthogonal. By doing so, the results obtained with a TDNNF model are often better than previous TDNN-LSTM and BLSTM results, while being much faster to decode [18].

2.1.2. LANGUAGE MODEL

The language model calculates the likelihood of a sequence of words $P(W)$. It predicts the next word given the previous words. The prior probability $P(W)$ of a word sequence $W = w_1, w_2, ..., w_K$ is:

$$P(W) = \prod_{k=1}^{K} P(w_k | w_{k-1}, ..., w_1) \tag{2.4}$$

By approximating it using a first-order Markov chain, the word following the current word depends on the current word only, and this kind of model is named as bigram (2-gram) and the equation above can be further simplified as:

$$P(W) = \prod_{k=1}^{K} P(w_k | w_{k-1}) \tag{2.5}$$

The current word is determined by the N preceding words, and this kind of model is named n-gram. When $n = 3$, the prior probability is calculated as follows:

$$P(W) = \prod_{k=1}^{K} P(w_k | w_{k-2} w_{k-1}) \tag{2.6}$$

$$P(w_k | w_{k-2} w_{k-1}) = \frac{c(w_{k-2} w_{k-1} w_k)}{c(w_{k-2} w_{k-1})} \tag{2.7}$$

where $c(w_{k-2} w_{k-1} w_k)$ denotes the total number of occurrence of word sequence $w_{k-2} w_{k-1} w_k$ in the dataset used, $c(w_{k-2} w_{k-1})$ corresponds to the occurrence of $w_{k-2} w_{k-1}$. For tri-gram (3-gram) or other n-gram models, the corpus may still have weird word combinations, so the data sparsity problem usually arises. Smoothing or pruning are mainly employed as the solution, which helps avoiding the probability of occurrence of a sequence of words being zero.

2.1.3. LEXICON

The most basic form of a lexicon is a set of words, followed by their pronunciation separated into units of word pronunciation, i.e. the phones. As its name suggests, it can be viewed as a dictionary for the dataset.

The lexicon bridges the acoustic model and the language model. As introduced above, the AM is defined via connecting the HMM of pronunciations and the feature vectors extracted from the speech recordings (MFCCs in our case). As we denote the acoustic feature be X, the AM models $P(X|W)$, the likelihood of an observable feature given a particular word. The LM estimates the prior probability $P(W)$ of a word in the transcripts. Between these two models, the lexicon maps the words from the LM to the features in the AM. By using AM and LM models in conjunction, the decoding procedure can be executed employing algorithms like Baum-Welch and Viterbi to compute the posterior probability $P(W|X)$, which represents the likelihood of a sequence of words given the features of speech signals. Speech recognition is done through picking the word sequence with the highest probability.

2.2. DATA AUGMENTATION

Data augmentation attempts to synthetically produce extra training data with a closer match to the target speaker, by transforming the original training data [3]. It has been proven to be an effective way to decrease the acoustic mismatch between training and testing conditions, since data augmentation approaches supplement the training data with distorted or synthetic variants of speech, with characteristics resembling the target acoustic characteristics, for instance a slower/faster speaking speed, a lower/higher volume or pitch. Other popular data augmentation techniques like SpecAugment were not tried out since they mostly only make changes to the spectrogram of original speech without increasing the amount of data.

By re-scaling the speed of the speech recordings in the time domain with a perturbation factor, both the audio duration and the spectral envelope are changed via speed perturbation [13]. When the value of the perturbation factor is bigger than 1 then the recording will be accelerated. When the value of perturbation factor is smaller than 1 the recording will be decelerated. Similarly, volume perturbation [13] re-scales the volume of the audio segments.

The pitch shift technique allows the original pitch of a sound to be raised or lowered [4]. As the features are extracted in the frequency domain, recordings with higher frequency (Hz) produce MFCCs of better quality [2].

2.3. TRAINING STRATEGIES

Based on the availability of labels in data sets, the transfer learning techniques can be categorised as shown in Table 2.1.

Since the Dutch data sets used in this research comprises labelled data only, fine-tuning and multi-task learning are naturally adopted as my training methods.

		Source Data	
		labelled	*unlabelled*
Target Data	*labelled*	Fine-tuning Multi-task Learning	Self-taught Learning
	unlabelled	Domain adversarial training Zero-shot Learning	Self-taught Clustering

Table 2.1: Categorisation of transfer learning techniques

2.3.1. FINE-TUNING

Fine-tuning is a sub-category of transfer learning, which means taking the weights of a trained neural network and using it as initialization for a new model being trained on new data. It is usually used to speed up the training, or overcome the sparse data problem since it's relatively efficient compared with training from scratch. There are various strategies within the scope of fine-tuning, such as training the whole initialized network or "freezing" some of the pre-trained weights (usually whole layers).

2.3.2. MULTI-TASK LEARNING

During multi-task learning, the system is trained for multiple tasks simultaneously using shared information. The idea is that this allows the model to exploit similarities and differences between the two tasks to create a model that is better able to generalise than models trained on a single task. Multi-task learning, where the secondary task is accent/dialect recognition, has been explored by a number of researchers [6, 21, 12] in the context of hybrid models, and improvements with multi-task learning have been observed in these research.

Multi-task learning aims to learn to produce generalized speech representations that are not too task-specific so that they can be be shared across different tasks. Through sharing knowledge, the data becomes more ample for each task. It also allows the model to learn representations transmitting enough knowledge for all of the tasks. An example of DNN with multi-task learning is displayed in Figure 2.8.

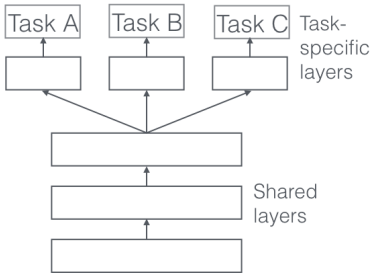

Figure 2.8: Multi-task Learning with shared layers

The task-specific parts of the network begin with the same representation from the last shared layer.

Multi-task learning improves the generalizability of this representation because learning multiple tasks forces the model to focus on the features that are useful across all of the tasks. Assuming the tasks are correlated, a feature that is important for Task A is also likely to be important for Task C. The opposite is also true; unimportant features are likely to be paid less attention by the system across all the tasks.

Multi-task learning also effectively increases the size of datasets, since the datasets for each task are shared. By adding more samples to the training set from different tasks, the model will learn to better ignore the task-specific noise or biases within each individual data-set.

The loss function for a multi-task learning model is as follows:

$$L_{MTL} = \lambda_1 L_{task1} + \lambda_2 L_{task_2} + ... + \lambda_n L_{task_n} \tag{2.8}$$

where n denoted the total number of tasks and all the λs sum up to 1.

2.4. ASR SYSTEM BUILT WITH KALDI

This section focuses on the implementation process of a hybrid ASR system using Kaldi toolkit. The baseline TDNNF model was built with Kaldi since it outperformed E2E model in terms of the absolute bias, so I will follow this and use Kaldi toolkit to build my model as well.

2.4.1. TRIPHONE

Speech is continuous. The pronunciation of a certain phone is influenced by the preceding and following phones, for example, the '/t/' sounds differently in 'suit' and 'tube'. Therefore, the acoustic phonetic context of a speech unit does affect its acoustic realization. As shown in Figure 2.9, 'speech' is labelled as $sil-s+piy-ch+sil$, where $-$ indicates that iy is followd by ch and $+$ means p follows s. This representation method which includes the context of a phone is called a triphone.

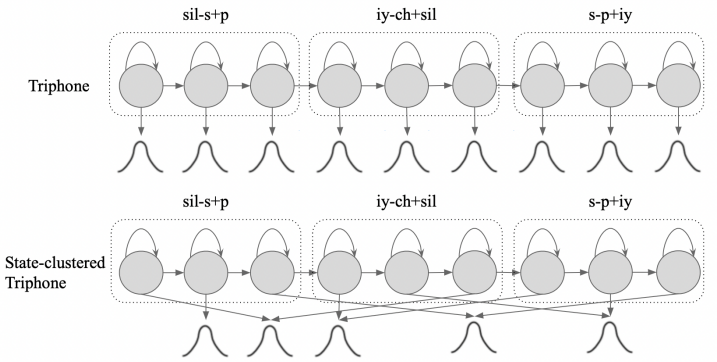

Figure 2.9: Triphone and state-clusered triphone

However, one issue with using triphones is that the number of states is exponentially increased during computation. In practice, many states have similar output distribution, so they can share the same Gaussian model, which is called state-clustering, as

we can see in Figure 2.9 as well. This clustering process can be efficiently implemented using phonetic decision trees, which are binary trees with a series of 'yes' or 'no' questions regarding the right and left context of every phone. As shown in Figure 2.10, for each monophone, a decision tree will be built involving an algorithm that aims local optimum: the algorithm picks the question that allows the data to be split resulting in the highest likelihood under the HMMs. Note that in Figure 2.10, the square box denote HMM, i.e. the cluster of triphones.

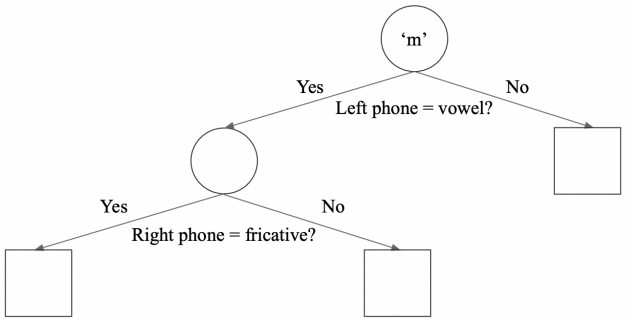

Figure 2.10: How decision tree splits data[15]

A triphone model built with this fashion was the very first step of the experiments.

2.4.2. WEIGHTED FINITE STATE TRANSDUCER (WFST)

Kaldi-based ASR system is framed in a WFST context, where each ASR component corresponds to a transducer, and the 'weights' denote transition probabilities from input to output of the transducer. The transducers used in a Kaldi-based ASR system are shown in table 2.2. The H transducer maps hidden states of an HMM to context-dependent

Transducer	Description	input	output
G	word-level grammar	words	words
L	pronunciation lexicon	phones	words
C	context-dependency	CD phones	phones
H	HMM	HMM states	CD phones

Table 2.2: Transducers used in a Kaldi-based ASR system

(CD) phones and C maps CD phones to context-independent phones, afterwards the L and G transducers map phones to words and then to sentences. The overall combined transducer $H \circ C \circ L \circ G$ represents the mapping from HMM states to word sequences restricted by G.

This process can be expressed as:

$$HCLG = min(det(H \circ min(det(C \circ min(detL \circ G))))) \tag{2.9}$$

With the equation above, the decoding can be done in an efficient way.

2.4.3. WORD LATTICE

When decoding a traditional GMM-HMM model, word lattices are required. A lattice in Kaldi refers to a representation of the alternative word-sequences that are "sufficiently likely" for a particular utterance, and is used as a data-structure frame and saves the N-best sequence paths with a more compact form to deal with the large dimensional search problem. Suppose an utterance with T frames will be decoded, we construct an acceptor named U, which has $T+1$ states with an arc for each combination of time and context-dependent HMM state. Then the search graph (S) of the utterance is defined as:

$$S = U \circ HCLG \tag{2.10}$$

The decoding problem is equal to finding the best path through S. In practice, Viterbi decoding using maximum likelihood (ML) is used to find the best path.

The final pruned lattice is set as P and its inverse is $Q = inv(P)$. Another acceptor is denoted by E, whose symbols equal to the input symbol (word) on the corresponding arc of Q, and the weights on the arcs of E containing both the weight and the output symbol (p.d.f.), if any, on the corresponding arcs of Q. Here E can be regarded as an encoded version of Q. The generated lattice L is:

$$L = prune(det(rmeps(E)), \alpha) \tag{2.11}$$

We obtained HMM state-level alignment information via determinization. Through pruning, only the best-scoring path for each word sequence is retained.

2.4.4. EVALUATION CRITERIA

WORD ERROR RATE

Automatic speech recognition performance is typically evaluated using the WER. The WER is exactly the Levenshtein distance between the prediction and the ground truth, i.e. the minimum number of single-character edits (insertions, deletions, or substitutions) required to change the prediction into the true sentence. WER is computed as:

$$WER = \frac{S + D + I}{N} \tag{2.12}$$

Where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the number of words in the reference transcript. Figure 2.11 gives an example of how WER is calculated. A lower WER indicates that the prediction generated appears to be more accurate.

Although WER is used widely as the mainstream metric for evaluating the performance of ASR systems, the drawback still exists: it cannot differentiate between important words and words that are not essential to the sentences.

BIAS

In this research, the bias against non-native accents is quantified as the absolute WER difference between the native speech data and the non-native accented speech data tested with the same ASR model. It can be expressed as:

$$Bias = |WER_{non-native} - WER_{native}| \tag{2.13}$$

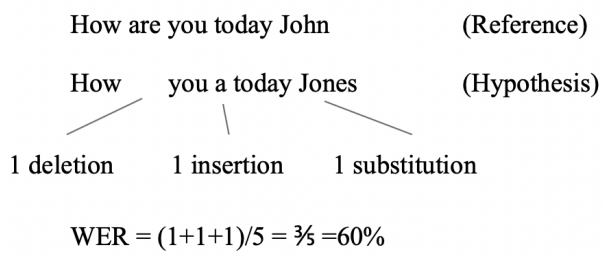

Figure 2.11: An example of WER

BIBLIOGRAPHY

[1] Ossama Abdel-Hamid et al. "Convolutional Neural Networks for Speech Recognition". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22 (2014), pp. 1533–1545.

[2] Mohammad Abushariah and Majdi Sawalha. "The effects of speakers' gender, age, and region on overall performance of Arabic automatic speech recognition systems using the phonetically rich and balanced Modern Standard Arabic speech corpus". In: Jan. 2013.

[3] Peter Bell et al. "Adaptation Algorithms for Neural Network-Based Speech Recognition: An Overview". In: *IEEE Open Journal of Signal Processing* 2 (2021), pp. 33–66. ISSN: 2644-1322. DOI: 10.1109/ojsp.2020.3045349. URL: http://dx.doi.org/10.1109/OJSP.2020.3045349.

[4] C. Bellettini and G. Mazzini. "Reliable Automatic Recognition for Pitch-Shifted Audio". In: *Proceedings of 17th International Conference on Computer Communications and Networks*. 2008, pp. 1–6. DOI: 10.1109/ICCCN.2008.ECP.157.

[5] Kai Chen and Qiang Huo. "Training Deep Bidirectional LSTM Acoustic Model for LVCSR by a Context-Sensitive-Chunk BPTT Approach". In: *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 24.7 (July 2016), pp. 1185–1193. ISSN: 2329-9290.

[6] Mohamed G. Elfeky et al. "Towards acoustic model unification across dialects". In: *2016 IEEE Spoken Language Technology Workshop (SLT)* (2016), pp. 624–628.

[7] Siyuan Feng et al. *Quantifying Bias in Automatic Speech Recognition*. 2021. DOI: 10.48550/ARXIV.2103.15122. URL: https://arxiv.org/abs/2103.15122.

[8] Mark Gales and Steve Young. "The Application of Hidden Markov Models in Speech Recognition". In: *Foundations and Trends® in Signal Processing* 1.3 (2008), pp. 195–304. ISSN: 1932-8346. DOI: 10.1561/2000000004. URL: http://dx.doi.org/10.1561/2000000004.

[9] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. "Hybrid speech recognition with Deep Bidirectional LSTM". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding* (2013), pp. 273–278.

[10] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. "Speech recognition with deep recurrent neural networks". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), pp. 6645–6649.

[11] Tong He et al. "Do Deep Neural Networks Outperform Kernel Regression for Functional Connectivity Prediction of Behavior?" In: Nov. 2018. DOI: 10.1101/473603.

[12] Abhinav Jain, Minali Upreti, and Preethi Jyothi. "Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning". In: *INTERSPEECH*. 2018.

[13] Tom Ko et al. “Audio augmentation for speech recognition”. In: *Proc. INTERSPEECH*. 2015, pp. 3586–3589. DOI: 10.21437/Interspeech.2015-711.

[14] Yann Lecun and Yoshua Bengio. “Convolutional networks for images, speech, and time-series”. English (US). In: *The handbook of brain theory and neural networks*. Ed. by M.A. Arbib. MIT Press, 1995.

[15] et.al Michael Picheny Bhuvana Ramabhadran. URL: http://www.ee.columbia.edu/stanchen/spring16/e6870/slides/lecture6.pdf.

[16] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. “A time delay neural network architecture for efficient modeling of long temporal contexts”. In: *INTERSPEECH*. 2015.

[17] Vijayaditya Peddinti et al. “Low Latency Acoustic Modeling Using Temporal Convolution and LSTMs”. In: *IEEE Signal Processing Letters* 25.3 (2018), pp. 373–377. DOI: 10.1109/LSP.2017.2723507.

[18] Daniel Povey et al. “Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks”. In: *Proc. Interspeech 2018*. 2018, pp. 3743–3747. DOI: 10.21437/Interspeech.2018-1417.

[19] A. Waibel et al. “Phoneme recognition using time-delay neural networks”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.3 (1989), pp. 328–339. DOI: 10.1109/29.21701.

[20] Laurens van der Werff. *laurensw75/kaldi_egs_CGN*. URL: https://github.com/laurensw75/kaldi_egs_CGN.

[21] Bishan Yang and Claire Cardie. “Joint Modeling of Opinion Expression Extraction and Attribute Classification”. In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 505–516. DOI: 10.1162/tacl_a_00199. URL: https://aclanthology.org/Q14-1039.

3

METHODOLOGY

In this chapter, the datasets, the data augmentation techniques, and the training methods used in this research are explained in further detail.

3.1. DATASETS

3.1.1. THE SPOKEN DUTCH CORPUS

CGN[4], which abbreviates from Corpus Gesproken Nederlands, is a Dutch corpus containing native speech data spoken by speakers from the Netherlands and Flanders. The size of the corpus is close to ten million words (about 1,000 hours of speech), two thirds of which originates from the Netherlands and one third from Flanders. The data recorded in only the Netherlands are used to train the ASR systems. CGN is a dataset of contemporary standard monologue and multilogue Dutch as spoken by adults (age 18-approximately 60 years) in The Netherlands and Flanders, which covers different speaking styles including read, broadcast news (BN), and conversational telephone speech (CTS).

The CGN corpus is further divided into 15 different speech data components as shown in the list below. Among the 15 components, components $a - h$ are multilogue speech data, while components $i - o$ are monologue speech data.

- Component a: face-to-face spontaneous conversations,
- Component b: interviews with teachers of Dutch,
- Component c: spontaneous telephone dialogues recorded via a switchboard,
- Component d: spontaneous telephone dialogues recorded with local interface,
- Component e: simulated business negotiations,
- Component f : interviews/discussions/debates (broadcast),
- Component g: (political) discussions/debates/meetings,

- Component h: lessons recorded in a classroom,
- Component i: live commentaries (broadcast),
- Component j: newsreports/reportages (broadcast),
- Component k: news (broadcast),
- Component l: commentaries/columns/reviews (broadcast),
- Component m: ceremonious speeches/sermons,
- Component n: lectures/seminars,
- Component o: read speech.

All the components above add up to 483 hours of speech from Netherlands in total duration. The pre-processing of the CGN data follows the recipe by [5], which segments the audio clips into smaller snippets of at least 6 seconds in duration and then removes the silent parts which are regarded as meaningless, leading to a final actual data set with 423-hour speech recordings, which is denoted by C_{train}. How the training set was picked also follows what has been done in [5].

3.1.2. JASMIN-CGN

As an extension of the CGN corpus, JASMIN-CGN[1] consists of read speech and human-machine interaction (HMI) speech spoken by native speakers who are children, teenagers and older adults and non-native speakers who are teenagers and adults. The non-native speakers come from 37 different countries, including Afghanistan, Andorra, Egypt and Spain. Same as how CGN is used, native speech from only Netherlands is used. The general information about these 5 speakers groups in JASMIN-CGN corpus is listed below.

- DC: native Dutch children; age 6-13; 12 hours 21 minutes of raw speech data,
- DT: native Dutch teenagers; age 12-18; 12 hours 21 minutes of raw speech data,
- DOA native Dutch older adults; age greater than or equal to 59, 9 hours 26 minutes of raw speech data,
- NNT: non-native teenagers; age 11-18; 12 hours 21 minutes of raw speech data,
- NNA non-native adults; age 19-55; 12 hours 21 minutes of raw speech data.

Following the same recipe [5] introduced in the above section, the recordings in JASMIN-CGN are pre-processed, resulting in a cleaned data set of 36.12 hours in duration. The non-native test sets are handpicked from the corpus to make it as fair as possible - by 'fair' I mean the test set consists of speakers who are native and non-native, male and female, children, teenagers, and older adults, each with an equal portion. For each group of speakers (DC, DT, DOA, NNT and NNA), 6 speakers (3 female and 3 male speakers) who record both read and HMI speech data are selected. What's left in JASMIN-CGN after picking out the test sets is used as the training set.

The test sets are listed below.

- R_D: native Dutch read speech; 1.45 hours; consisting of R_{DC}, R_{DT} and R_{DOA},
- R_{NN}: non-native read speech; 1.63 hours; consisting of R_{NNT} and R_{NNA},
- H_D: native HMI speech; 0.68 hours; consisting of H_{DC}, H_{DT} and H_{DOA},
- H_{NN}: non-native accented HMI speech; 0.36 hours; consisting of H_{NNT} and H_{NNA}.

3.2. BASELINE STATE-OF-THE-ART ASR SYSTEM FOR DUTCH

Starting with a monophone system trained with Kaldi, the initial HMM topology was created. An initial triphone model was first trained, then TDNNF-based acoustic models are built on top of this basic triphone model, replacing the GMM obtained from previous stages, but with the same HMM used as before.

The baseline model, the TDNNF hybrid DNN-HMM architecture built with Kaldi from [2], is illustrated in figure 3.1. The TDNNF model consisted of 12 TDNNF layers of dimension 1024, and was trained with the lattice-free maximum mutual information (LF-MMI) criterion for 4 epochs. 100-dimensional i-vectors were appended to the high resolution MFCC input features for speaker adaptation purposes. Context-dependent phone alignment labels used for training the AM were obtained by using a GMM-HMM trained beforehand with the same training data as that for the TDNNF. The baseline

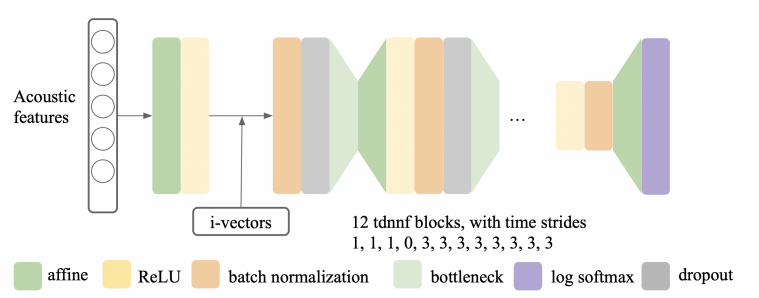

Figure 3.1: Architecture of the employed TDNNF AM.

model was trained with C_{train}.

3.3. DATA AUGMENTATION

Three different data augmentation techniques were investigated and compared. The data is augmented two-fold by each of the augmentation techniques, which generated 2 times the original data in terms of total length.

3.3.1. SPEED PERTURBATION

The standard Kaldi speed perturbation script that re-scales the speed of the speech recordings in the time domain is used. The perturbation factors we applied were $\{0.9, 1, 1.1\}$. Given an audio segment $x(t)$, the scaling factor α is applied along the time axis, giving the output $y(t)$ as follows:

$$y(t) = x(\alpha t) \tag{3.1}$$

In frequency domain, this corresponds to the change below:

$$X(f) -> \frac{1}{\alpha} X(\frac{1}{f}) \tag{3.2}$$

where $X(f)$ and $\frac{1}{\alpha} X(\frac{1}{f})$ represent the Fourier transform of $x(t)$ and $y(t)$ respectively. In this way, speed perturbation leads to both change in audio duration and perturbation in spectral envelope [3].

3.3.2. VOLUME PERTURBATION

Similar to speed perturbation, volume perturbation [3] re-scales the volume of the audio segments. We used the standard Kaldi script that modifies the wav.scp to perturb the volume. The same rescaling factors as used for speed perturbation are applied for volume perturbation, $\{0.9, 1, 1.1\}$.

3.3.3. PITCH SHIFT

The pitch shift technique allows the original pitch of a sound to be raised or lowered. In our work, which uses the *librosa* function $librosa.effects.pitch\_shift$, the pitches of audio snippets are shifted by $\{\pm 2\}$ semitones. A semitone corresponds to multiplying the number of Hertz (Hz) by $2^{\frac{1}{2}}$.

3.4. TRANSFER LEARNING

Two transfer learning techniques were investigated and compared to each other and standard training (referred to as in-domain training).

3.4.1. FINE-TUNING

Fine-tuning takes the initial baseline model trained on C_{train}, and then trains the new model with a target data set. The model is trained for four epochs, following the scheme used in the baseline. Layer transfer was employed during training, where the values of parameters are transferred from the baseline to be the initial values of the new model. During fine-tuning, the baseline Gaussian Mixture Model (GMM), i-vector extractor, tree, and TDNNF architecture are used, while the target training data and a fused tri-gram language model in which the word combinations and words from both Jamin-CGN and CGN are used, as Jasmin-CGN contains phones and words unseen in CGN.

3.4.2. MULTI-TASK LEARNING

Figure 3.2 shows how multi-task learning is implemented in the AM in our TDNNF-architecture.

During multi-task learning, the model is trained for recognition of the speech in CGN and recognition of the speech in Jasmin-CGN. The acoustic features and the acoustic model are shared except for the last hidden layer of the neural network in the AM. The features include 100-dimensional i-vectors extracted from the global i-vector extractor trained on both CGN and Jasmin-CGN and appended to the MFCC features. The AM is the TDNNF and the LM is the fused tri-gram language model in which the text and words from both Jamin-CGN and CGN are used.

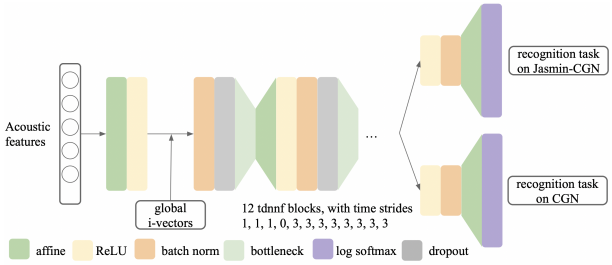

Figure 3.2: Multi-task Learning

The loss of the whole network is computed as a weighted sum of cross-entropy losses at two different output layers; they weigh the same (0.5) in our case.

3.5. EXPERIMENTS AND EVALUATION

In our experiments, the native Dutch speech from CGN C_{train} was used to train the initial baseline model after which a new model was trained using fine-tuning and multi-task learning.

Training data from the Jasmin-CGN and CGN is augmented and fed into the network simultaneously when doing in-domain training and multi-task learning, while fine-tuning is performed on the basis of an AM trained with CGN and its augmented data. Five sets of target data consisting of both native and non-native Dutch are created and used for fine-tuning, multi-task training and added to the training set for in-domain training:

- a) the original J_{train} speech data;
- b) speed perturbed speech from J_{train};
- c) volume perturbed speech from J_{train};
- d) pitch shifted speech from J_{train};
- e) b+c+d

Both the native and non-native accented speech from Jasmin-CGN is used for speech, volume and pitch shift augmentation. Thereafter the training is carried out in three ways as below for each of the above five data combinations:

- In-domain training: The model is trained on the CGN C_{train} and the Jasmin-CGN and the augmented data simultaneously [i.e., data a) to e)].
- Fine-tuning: The baseline model trained on C_{train} is fine-tuned with the Jasmin-CGN data and the augmented data [i.e., data a) to e)].
- Multi-Task Learning: The model is trained on the CGN C_{train} and the Jasmin-CGN and the augmented data simultaneously [i.e., data a) to e)]. The last hidden layer and output layer are independent per data set.

All 15 models are evaluated on the four defined test sets of JASMIN-CGN in terms of the word error rate (WER). Bias is defined as the difference in WER between the native Dutch speakers and the non-native Dutch speakers, and calculated for read speech and HMI speech separately.

3.6. DATA PREPARATION

The Kaldi recipes require some standard input files to build an ASR system, and these files can be prepared using the corpora. The data preparation stage mainly consists of processing of the recordings in the database, and processing of the lexicon, the phone set, and meta-data about the phone set which Kaldi needs. For acoustic data, the following files are needed:

- **text**

  The file "text" contains the transcriptions of each utterance with the first element represents the utterance-id. The format of this file is <utterance-id> <transcripts>:

N08006-fn008183.1	ggg
N08006-fn008183.10	ja
N08006-fn008183.103	ja
N08006-fn008183.104	ja
...	

- **wav.scp**

  The format of this file is <recording-id> <extended-filename>:

fn008053 sox -t wav ...data/audio/wav/comp-c/nl/fn008053.wav -b 16 -t wav - remix - |
fn008093 sox -t wav ...data/audio/wav/comp-c/nl/fn008093.wav -b 16 -t wav - remix - |
...

- **utt2spk**

  This file shows that for each utterance, which speaker spoke it. The format is <utterance-id> <speaker-id>:

N08006-fn008183.1	N08006
N08006-fn008183.10	N08006
N08006-fn008183.103	N08006
N08006-fn008183.104	N08006
...	

- **spk2utt**

  The format of spk2utt is <speaker-id> <utterance-id1> <utterance-id2> ....:

N08006 N08006-fn008183.1 N08006-fn008183.10 N08006-fn008183.103 ...

- **feats.scp**

  This file is related to the feature extraction process and the format is <utterance-id> <extended-filename-of-features>. In the following example, the second element means when opening the "archive" file .../egs/kaldi_egs_CGN/s5/data/dev_t/data/raw_ fseek() to position 18, and the computer reads the data stored there.

N08006-fn008183.1 .../egs/kaldi_egs_CGN/s5/data/dev_t/data/raw_mfcc_dev_t.1.ark:18
N08006-fn008183.10 .../egs/kaldi_egs_CGN/s5/data/dev_t/data/raw_mfcc_dev_t.1.ark:1930
N08006-fn008183.103 .../egs/kaldi_egs_CGN/s5/data/dev_t/data/raw_mfcc_dev_t.1.ark:2296
...

These files map the information of certain speech segment to the corresponding speaker.

The second group of data files are for the language model. The essential LM-related files are:

- **lexicon.txt**

  The lexicon.txt file for both CGN and Jasmin-CGN are modified from the lexicon given by each database. Only orthographic and phonetic information are extracted. For JASMIN-CGN, silences are denoted by []. The repeated entries for the same word on multiple lines are removed from lexicon. An example of file lexicon.txt is as follows:

aalmoezenier	a l m u z @ n i r
aalmoezeniersdienst	a l m u z @ n i r z d i n s t
aalscholver	a l s x O l v @ r
aalscholvers	a l s x O l v @ r s
...	

  The first element is the word and the following items are the corresponding transcriptions in terms of phonemes.

- **nonsilence_phones.txt**

  This file includes all phones listed in the transcripts. Different duration-dependent versions of the same phone are expressed using the ′ :′ symbol:

- **silence_phones.txt & optional_silence.txt**

  silence_phones.txt contains two markers SIL and SPN which are abbreviation of "silence word" and "spoken noise" respectively. SPN is linked together with <UNK>, which means unknown phones. Kaldi maps all words that appear in training data but not in the lexicon to <UNK>. Similarly, the file optional_silence.txt only contains SIL.

2
2:
@
@:
...

When viewing both Jasmin-CGN and CGN as in-domain data, the text files above are merged together as a whole.

- **extra_questions.txt**

  Additional questions used when generating decision tree during state-clustering are listed in this file. These questions are used during data splitting.

  The final step of preparing data is generating a WFST form for the grammar(G transducer) and lexicon(L transducer) respectively using the language files obtained.

BIBLIOGRAPHY

[1] Catia Cucchiarini et al. "JASMIN-CGN: Extension of the Spoken Dutch Corpus with Speech of Elderly People, Children and Non-natives in the Human-Machine Interaction Modality". In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA), May 2006. URL: http://www.lrec-conf.org/proceedings/lrec2006/pdf/254_pdf.pdf.

[2] Siyuan Feng et al. *Quantifying Bias in Automatic Speech Recognition*. 2021. DOI: 10.48550/ARXIV.2103.15122. URL: https://arxiv.org/abs/2103.15122.

[3] Tom Ko et al. "Audio augmentation for speech recognition". In: *Proc. INTERSPEECH*. 2015, pp. 3586–3589. DOI: 10.21437/Interspeech.2015-711.

[4] Nelleke Oostdijk. "The Spoken Dutch Corpus. Overview and First Evaluation". In: *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*. Athens, Greece: European Language Resources Association (ELRA), May 2000. URL: http://www.lrec-conf.org/proceedings/lrec2000/pdf/110.pdf.

[5] Laurens van der Werff. *laurensw75/kaldi_egs_CGN*. URL: https://github.com/laurensw75/kaldi_egs_CGN.

4

RESULTS

Several hybrid TDNNF DNN-HMM architectures are trained using different sets of augmented data obtained with the three data augmentation techniques on the basis of the baseline hybrid TDNNF DNN-HMM. Model performance is analysed in comparison with the baseline results on native and non-native accented Dutch.

4.1. RESULTS

Our baseline ASR system was first evaluated on the CGN test sets to investigate its performance on in-domain data, making sure that it recognises native Dutch speech well. Subsequently, we investigate the new models' performance on the Jasmin-CGN test sets.

4.1.1. BASELINE RESULTS

CGN does not have an HMI set, so instead we tested the baseline on a similar speaking style: conversational telephone speech. Table 4.1 shows the recognition and bias results of the baseline model, trained only on the CGN native speech, for read speech and conversational speech separately. Average Dutch (AvgD refers to the WER averaged over all the native Dutch speakers and Av(era)g(e)N(on-native) speakers refers to the WER averaged over all the non-native Dutch speakers.

Read speech is better recognised than conversational speech. At the same time, the bias against non-native listeners is about twice as large for read speech than conversational speech.

4.1.2. DATA AUGMENTATION AND TRANSFER LEARNING RESULTS

Table 4.2 shows the recognition results for the native and non-native speech and the bias results for read speech (B_R) and HMI speech (B_H) separately for the three training methods with the five different data augmented training sets. The WER is averaged over all native Dutch speakers and all non-native Dutch speakers respectively on read speech and HMI speech.

For in-domain training, we observe that the different data augmentation techniques when applied alone give little to no improvement in recognition performance for both the native and non-native speakers. Applying all three data augmentation techniques, however, leads to a reduction in WER and the lowest WER for both the native and the non-native speaker groups for both read speech and HMI speech. The lowest bias for HMI is also obtained when using all three data augmentation techniques, while for read speech the lowest bias was observed when volume perturbed data was added.

When fine-tuning is applied we observe a similar trend as for in-domain training, different results when different data augmentation techniques are applied alone, with the best recognition results obtained when all three data augmentation techniques are applied for both native and non-native speakers and both read and HMI speech. The smallest bias for read speech was observed when only the Jasmin data was added, for HMI speech the smallest bias was observed when all augmented data was added during fine-tuning.

For multi-task training, the smallest bias for read speech is also the smallest bias overall, which is obtained with pitch-shifted Jasmin data. The smallest bias in HMI speech comes from training with speed-perturbed Jasmin.

Among the techniques employed, fine-tuning and multi-task learning reduce the bias more than simply including the target non-native speech as in-domain data, although the imbalanced nature of Jasmin-CGN, which refers to its uneven composition where the amount of HMI speech data is more than read speech data, made the results of fine-tuning overfit to the HMI speech.

Looking at the variation of WERs across speaker groups, it is true that data augmentation helps reduce the WER, but it is not definite that the application of data augmentation technique(s) reduces the bias, e.g. B_R becomes higher than what obtained in in-domain training when Jasmin-CGN is augmented during fine-tuning, whereas B_H is reduced.

If we look at the effect of each data augmentation technique individually on recognition performance and bias, it is clear that in most cases pitch shift gives the most bias reduction whether it is tested on read speech or HMI speech. Speed perturbation ranks the second on average and volume perturbation comes last, suggesting that non-native speakers do not vary much from the native speakers in case of speech volume. When only pitch shift is applied, the bias in HMI speech is usually smaller than the bias when volume or speed perturbed data is used. Furthermore, multi-task learning achieved the smallest bias in read speech with only pitch shift applied.

Comparing the different training methods shows that multi-task learning gives the best performance as its bias is lower in most datasets than in-domain training and fine-tuning.

In general, read speech is better recognised that HMI speech, which is true for all speaker groups irrespective of the data augmentation techniques applied. B_R is the largest when all the augmented data is used during fine-tuning, and the smallest when Jasmin-CGN is only pitch-shifted during multi-task learning. B_H is the largest when we do the in-domain training with the original datasets, and the smallest when speed perturbation is applied during multi-task learning.

B_R is the largest when all the augmented data is used during fine-tuning, and the

smallest when Jasmin-CGN is only pitch-shifted during multi-task learning. B_H is the largest when we do the in-domain training with the original datasets, and the smallest when speed perturbation is applied during multi-task learning.

Table 4.1: WERs(%) on the read and HMI native/non-native speech. Baseline model trained with CGN only.

Group	AvgD	AvgN	Bias
Read	20.80	48.04	27.24
HMI	30.90	44.57	13.67

Table 4.2: WERs(%) on the read and HMI native/non-native speech for the models trained with different training methods (in-domain, fine-tuning, and multi-task) and different types of augmented speech data. SP refers to speed perturbation; VP refers to volume perturbation; PS refers to pitch shift. Column-wise, the lowest WER and bias are denoted in bold.

Method	Datasets	R_D	R_{NN}	H_D	H_{NN}	B_R	B_H
in-domain	C_{train}, J_{train}	17.97	31.65	28.8	37.95	13.68	9.15
	$C_{train}, J_{train} + SP$	17.55	30.13	29.47	36.65	12.58	7.18
	$C_{train}, J_{train} + VP$	20.49	32.54	29.9	37.65	12.05	7.75
	$C_{train}, J_{train} + PS$	17.26	30.04	28.59	36.33	12.78	7.74
	$C_{train}, J_{train} + SP + VP + PS$	16.82	30.04	27.95	**34.66**	13.22	6.71
fine-tune	J_{train}	15.61	31.09	45.24	53.7e	15.48	8.48
	$J_{train} + SP$	15.31	30.89	45.1	52.81	15.58	7.71
	$J_{train} + VP$	15.66	31.45	46.46	53.96	15.79	7.5
	$J_{train} + PS$	13.85	30.3	47.06	54.55	16.45	7.49
	$J_{train} + SP + VP + PS$	**12.64**	29.91	43.79	50.1	17.27	**6.31**
multi-task	C_{train}, J_{train}	21.11	34.8	29.05	35.98	13.69	6.93
	$C_{train}, J_{train} + SP$	20.03	34.05	28.67	35.37	14.02	6.7
	$C_{train}, J_{train} + VP$	20.84	33.73	29.01	35.86	12.89	6.85
	$C_{train}, J_{train} + PS$	18.79	27.88	28.29	35.06	**9.09**	6.77
	$C_{train}, J_{train} + SP + VP + PS$	17.05	**27.87**	**28.03**	34.99	10.82	6.96

5

Discussion and Conclusion

5.1. Discussion

Among the data augmentation techniques adopted, pitch shift contributed the most to the overall reduction in bias. A possible explanation could be that, compared to speaking volume and speaking speed, the pitch difference between native and non-native speakers gives more variation to the speech data within a dataset. Another noticeable finding observed from the table is that combining all data augmentation techniques does not necessarily lead to better performance in terms of bias reduction, as sometimes training with only one set of augmented data has lower bias. As for the effect of transfer learning, the results show that the application of fine-tuning makes the model work better for read speech than the HMI data. One possible reason could be that there are more read speech than HMI speech in JASMIN-CGN, . Hence, the model fine-tuned with JASMIN-CGN is slightly biased towards read speech.

On the other hand, multi-task learning has managed to avoid this kind of performance degradation. Furthermore, multitask learning enforces more fairness across native/non-native speaker groups than fine-tuning, as most biases are lower in multitask learning than those obtained in fine-tuning.

The research questions can be answered.

- RQ1: Could data augmentation help reduce the bias against non-native accented speech in ASR systems?

- A1: Yes. As shown in Table 4.2 for in-domain training the results on adding Jasmin-CGN and the perturbed data with different augmentation techniques, decreases the WER and Bias significantly as compared to the baseline. Within the in-domain experiments, we observe that the different data augmentation techniques when applied alone give only little improvement in recognition performance for both the native and non-native speakers (and a small deterioration when only volume perturbed data is applied). Applying all three data augmentation techniques, however, leads to a reduction in WER and the lowest WER for both the native and the

non-native speaker groups for both read speech and HMI speech. The lowest bias for HMI is also obtained when using all three data augmentation techniques, while for read speech the lowest bias was observed when volume perturbed data was added, but this is due to an increase in WER for the native speech which was larger than the increase in WER for the non-native accented speech. In general, read speech is better recognised that HMI speech, which is true for all speaker groups irrespective of the data augmentation techniques applied.

- RQ2: Could fine-tuning and multi-task learning be effective in reducing bias against non-native accented speech when compared with standard training methods?

- A2: Yes. When fine-tuning is applied we observe a similar trend as for in-domain training, different results when different data augmentation techniques are applied alone (with an increase in WER when only volume perturbed data is applied), with the best recognition results obtained when all three data augmentation techniques are applied for both native and non-native speakers and both read and HMI speech. The smallest bias for read speech was observed when only the Jasmin data was added, but again at the cost of high WERs for both speaker groups and both types of speech. For HMI speech, the smallest bias was observed when all augmented data was added during fine-tuning.

  For multi-task training, the smallest bias for read speech is also the smallest bias overall, which is obtained with pitch-shifted Jasmin-CGN data. The smallest bias in HMI speech comes from training with speed-perturbed Jasmin-CGN.

  Among the techniques employed, fine-tuning and multi-task learning reduce the bias more than simply including the target non-native speech as in-domain data.

By comparing the best performance of each method, we can conclude that data augmentation does contributed to the reduction of both WER and bias, and among all the data augmentation techniques adopted, pitch shift is proven the most effective in most cases. The application of transfer learning methods, fine-tuning and multi-task learning, leads to better performance than what we got from simply using in-domain training. Furthermore, the lowest bias does not necessarily correspond to the lowest WER: for read speech, the model with the lowest bias has fairly good WERs, while for HMI speech, the WERs of the model with the lowest bias are relatively high. Looking at the WERs across different methods, we can see that multi-task learning shows the lowest bias for most datasets compared to in-domain training and fine-tuning. Multi-task learning reduces the bias while causing the least harm to the WERs among the methods adopted, but at the cost of slightly higher WERs on native speakers compared with fine-tuning. Overall, multi-task learning with pitch-shifted data seems to be the best choice if we aim to reduce the bias without causing performance degradation on native speakers.

5.2. FUTURE RESEARCH

Future-work-wise, the research could be done following the pattern of answering the research questions. More powerful data augmentation techniques could be explored to cover other features of non-native speech, such as articulation imprecision, disfluency,

and uncommon word combinations. Generative adversarial networks (GANs)[2] and Text-to-speech (TTS)[1] have proven effective in terms of augmenting original data to compensate for the lack of accented data, the possibility of generating more non-native speech to reduce the bias using these two techniques combined does exist.

On the other hand, effective learning methods not only limited to fine-tuning and multi-task learning could also be tried out, like the pre-training method of language models using corpus from other languages. Unlabelled data could also be exploited to help enhance the performance, since in this research, only experiments on labelled data are conducted, which limits the choices of adding possible extra datasets. Also, Using more task-specific layers during transfer learning could also be a possible approach, since the speech characteristics preserved in each layer could be very different.

5.3. CONCLUSION

In this research, it has been shown that although the bias against non-native speakers cannot be removed completely, the combination of certain techniques does help reducing it. The results show that the application of data augmentation techniques reduces bias against non-native-accented speech of HMI speech more than it reduces the bias for read speech. This suggests that the recognition accuracy of the ASR system is more sensitive to the change in HMI speech data. Both transfer learning methods adopted have promising results in terms of bias reduction, where multitask learning has a stronger effect compared with fine-tuning.

BIBLIOGRAPHY

[1] Alëna Aksënova et al. “Accented Speech Recognition: Benchmarking, Pre-training, and Diverse Data”. In: May 2022. DOI: 10.48550/arXiv.2205.08014.

[2] Xian Shi et al. *The Accented English Speech Recognition Challenge 2020: Open Datasets, Tracks, Baselines, Results and Methods*. 2021. arXiv: 2102.10233 [cs.SD].